2

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

ADAPTIVE DIM POINT TARGET DETECTION AND TRACKING IN INFRARED IMAGES

by

Thomas V. DeMars, Jr.

December 1988

Thesis Advisor: Charles W. Therrien

Approved for public release; distribution is unlimited.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION NAVAL POSTGRADUATE SCHOOL	6b OFFICE SYMBOL (If applicable) 54
7a NAME OF MONITORING ORGANIZATION NAVAL POSTGRADUATE SCHOOL	
6c ADDRESS (City, State, and ZIP Code) MONTEREY, CA 93943-5000	7b ADDRESS (City, State, and ZIP Code) MONTEREY, CA 93943-5000
8a NAME OF FUNDING/SPONSORING ORGANIZATION	8b OFFICE SYMBOL (If applicable)
9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER	
8c ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)
ADAPTIVE DIM POINT TARGET DETECTION AND TRACKING IN INFRARED IMAGES

12 PERSONAL AUTHOR(S)
DeMARS, THOMAS V., Jr.

13a TYPE OF REPORT MASTER's THESIS	13b TIME COVERED FROM ________ TO ________	14 DATE OF REPORT (Year, Month, Day) 1988, DECEMBER	15 PAGE COUNT 148

16 SUPPLEMENTARY NOTATION THE VIEWS EXPRESSED IN THIS THESIS ARE THOSE OF THE AUTHOR AND DO NOT REFLECT THE OFFICIAL POLICY OR POSITION OF THE DEPARTMENT OF DEFENSE OR THE U.S. GOVERNMENT.

17 COSATI CODES

FIELD	GROUP	SUB-GROUP

18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
TWO-DIMENSIONAL LMS, TWO-DIMENSIONAL RLS, SIGNIFICANCE TESTING, POINT TARGET DETECTION AND TRACKING, IMAGE PROCESSING, TWO-DIMENSIONAL PREDICTION

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

The thesis deals with the detection and tracking of dim point targets in infrared images. Research topics include image process modeling with adaptive two-dimensional Least Mean Square (LMS) and Recursive Least Squares (RLS) prediction filters. Target detection is performed by significance testing the prediction error residual. A pulse tracker is developed which may be adjusted to discriminate target dynamics. The methods are applicable to detection and tracking in other spectral bands.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a NAME OF RESPONSIBLE INDIVIDUAL CHARLES W. THERRIEN	22b TELEPHONE (Include Area Code) (408) 646-3347
22c OFFICE SYMBOL 62Ti	

DD FORM 1473, 84 MAR   83 APR edition may be used until exhausted
All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

☆ U.S. Government Printing Office: 1986—606-24.

UNCLASSIFIED

ADAPTIVE DIM POINT TARGET DETECTION AND TRACKING IN INFRARED IMAGES

by

Thomas V. DeMars, Jr.
Major, United States Marine Corps
B.S., Colorado School of Mines, 1976

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1988

Author: Thomas V DeMars Jr.
Thomas V. DeMars, Jr.

Approved by: Charles W. Therrien
Charles W. Therrien, Thesis Advisor

Murali Tummala
Murali Tummala, Second Reader

John P Powers
John P. Powers, Chairman, Department of
Electrical and Computer Engineering

G E Schacher
Gordon E. Schacher, Dean of Science
and Engineering

ABSTRACT

The thesis deals with the detection and tracking of dim point targets in infrared images. Research topics include image process modeling with adaptive two-dimensional Least Mean Square (LMS) and Recursive Least Squares (RLS) prediction filters. Target detection is performed by significance testing the prediction error residual. A pulse tracker is developed which may be adjusted to discriminate target dynamics. The methods are applicable to detection and tracking in other spectral bands.

TABLE OF CONTENTS

ACKNOWLEDGMENT

This thesis is dedicated to my wife Euphama and my loving daughters Euphama, Jacqueline and Tiffany. Without their love, understanding and patience this thesis and my success in the MSEE program would not be possible. I also want to thank Dr. Charles Therrien for his support and perseverance as my thesis advisor.

I. INTRODUCTION

A. PROBLEM DEFINITION

In recent years considerable effort has been put into the improvement of infrared sensors. Increased sensor resolution was one of the more ardently pursued goals. The drive behind the desire for better resolution came from several areas. One of the driving forces was the military's need for infrared sensors that were capable of detecting heat signatured objects at long range.

The specific concern in this thesis is the infrared detection of heat-signatured objects with sensors at altitudes of several hundred kilometers. Specifically, the thesis will be concerned with the detection and tracking of airborne targets such as medium range missiles and aircraft. The detection and tracking problem will be addressed from a generalized infrared sensor viewpoint. This is done with the desire to produce methods with the widest applicability.

B. BACKGROUND

The signature size and velocity of the targets have considerable bearing on the direction taken in solving this problem. Typical sensor geometric resolutions are on the order of .08 mrad. [Refs. 1 through 5] This produces a footprint on the earth's surface of approximately 48 m for a sensor at 600 km (1 μrad=1 m at 1000 km). The type of targets considered in development of the problem have signatures smaller than the footprint of the sensor. This will be assumed to hold true even with the effects of point spreading and optical blur inherent in the sensors [Refs. 3 and 4]. Sensor sampling rates are typically on the order of 1 sample per resolution cell, here .08 mrad [Ref. 3]. The resultant intensity of a sample containing a target is due to the sum of target thermal intensity and the background intensity within the footprint [Ref. 2]. Thus the targets, with a modified intensity, in a discrete image are nominally about one pixel in dimension.

The velocity of the targets is considerably larger than the drift rate due to sensor motion (e.g., two percent of the detector footprint) and jitter [Ref. 4]. If the image scan rate is typically 30 image frames per second, normal target movement between images will be less than two pixels per frame. A majority of the targets considered would actually move at less than one pixel per frame.

Many of the infrared systems contain more than one focal plane assembly operating at different wavelengths. The resulting images can be dealt with as individual images each with its own statistical characterization. In a single focal plane the scenes may be characterized by several 2-D random processes. The transition from one process to an adjacent process may be correlated or uncorrelated. The processes describe such items as terrain, and multiple layers of clouds and have a mean value dependent on the scene and the spectral band. [Ref. 4] Additionally the effects due to atmospheric jitter, background scintillation, and electronic noise may be modeled by a combination of white and correlated gaussian noise.

C. CONCEPT AND CONTENT OF THE THESIS

The preceding sections have provided the background to develop a conceptual solution for the problem. It can be seen that ultimately the detection algorithm will need to detect pixel-size targets with intensities that may differ only slightly from the surroundings. In tracking, the primary region of operation is seen to involve target motion of one to two pixels between successive image frames.

In this thesis the solutions will be developed for images in a single focal plane. Their applicability will be general in nature and do not depend on the frequency band or the detailed sensor characteristics. Some ideas for merging the results from applying the algorithms in several focal planes simultaneously will be addressed in the final chapter.

The next four chapters deal with the detection of the target and Chapter VI deals with the tracking. Initially a method will be proposed for modelling the background processes present in the image (terrain, clouds, etc.). In subsequent developments the background processes will be assumed to be homogeneous. Surveys of the literature show that frequently 90% of the image may be characterized by a single random process. The modelling process will be performed with fixed length spatially adaptive two-dimensional prediction filters. Adaptive filters used in the modelling process are the topics of Chapters III and IV. In these chapters the two-dimensional Least Mean Square (LMS) and Recursive Least Squares (RLS) filters are derived and tested.

The detection process is the subject of Chapter V. Linear prediction produces an error residual process which is stored. This error process contains the unpredictable information in the image. In Chapter V a method based on statistical significance testing will be used to separate the error residual into levels of predictability. The error residual present at the target location will be shown to be anomalous by comparison. This results in highlighting the targets after thresholding to achieve a low constant false alarm rate.

In Chapter VI a point target tracker is developed and tested. The tracker is the based on a maximum a posteriori (MAP) estimate of the target position. In the overall system the detection algorithm passes its results to an autonomous tracking algorithm. Consideration of the logical interface between the detection algorithm and the tracking algorithm are addressed in Chapter VI. The precept throughout is the development of a set of algorithms that is reasonably generic. Algorithms developed extend readily to images other than infrared.

II. MODEL

A. GENERAL DESCRIPTION

In order to pursue the objectives presented in the previous chapter it is necessary to establish a framework from which to work. Throughout this thesis the image to be processed is taken to be comprised of three components. The three components maybe viewed as a background generation process, a corrupting noise process, and a target generation process. Since target detection is the ultimate objective, it will be necessary to separate the three components.

Background generation processes have been studied by many researchers; the result is a wide variety of models. If the image can be divided into statistically similar regions, then each region may be modeled on an individual basis. It will further be assumed that although the overall image may be nonstationary, the statistics in the separate regions are individually stationary. The region within the image is taken to be the result of a linear shift invariant operation on a wide sense stationary Gaussian distributed white noise source of variance σ_w^2. [Ref. 6] A block diagram of the 2-D linear operation is shown in Figure 2-1.

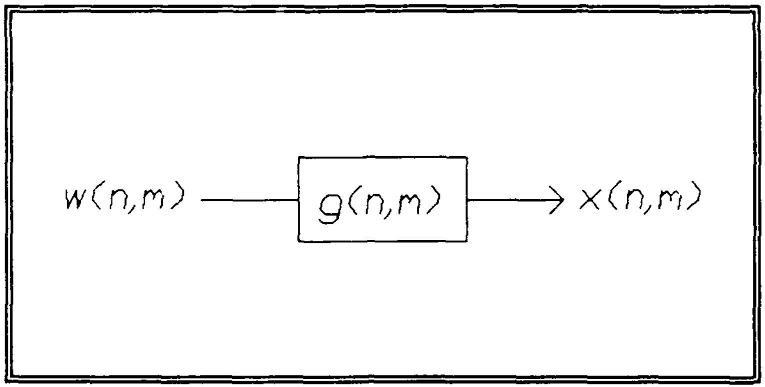

Figure 2-1. Block Diagram

The particular linear operation to be used is a two dimensional autoregression operation which may be written as:

$$x(n,m) = \sum_{\ell} \sum_{k} h(\ell,k)\ x(n-\ell,m-k) + w(n,m) \qquad (\ell,k) \in \beta \tag{2-1}$$

where β is some chosen region of support for the filter coefficients. This model has been termed the White Noise Driven Representation (WNDR) and the transfer function may be viewed as an all-pole IIR filter with a constant numerator driven by a white noise source. The filter is said to be "causual" (in the 2-D sense) if the computation of x(n,m) does not require knowledge of "future" values of the process. Otherwise it is said to be noncausal. The definition of causality is tied to the scan pattern used in processing the image. Figure 2-2 depicts the issue of causality for a row scan direction.

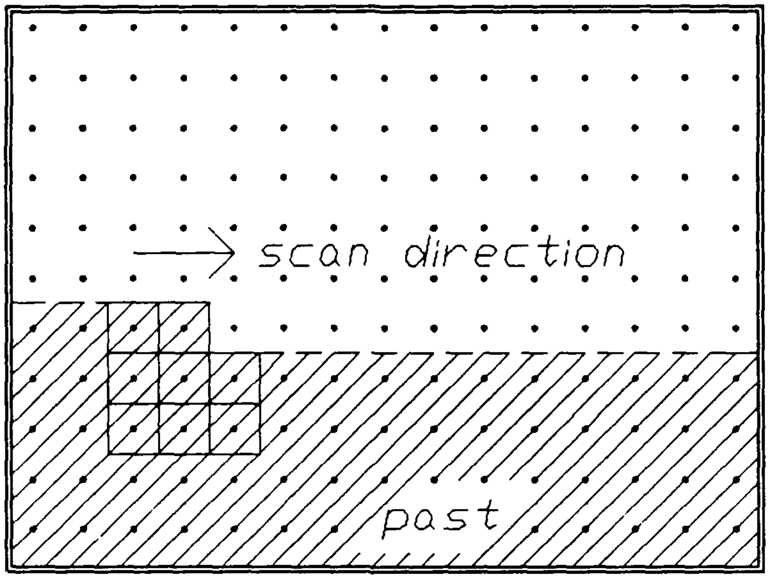

Figure 2-2. Casuality Depiction

If it is possible to determine the coefficients which describe the regional process, it then becomes possible to inverse filter the region. This will ideally leave only the driving noise process.

Noise present in the image is assumed to be additive although not necessarily gaussian or white. If the noise is colored, the same procedure used to identify and eliminate the regional process would be utilized on the noise. The result, as before, is a white noise process.

The most difficult of the three components of the image to characterize is the target process. If a specific model for the target is specified then the possibility exists that the overall system would be too restrictive. In order to avoid this pitfall it is proposed that the only identifying features of the target are its pixel/sub-pixel size, opaqueness, and non-zero mean.

B. PREDICTION ERROR FILTER

It has been shown [Refs. 6 and 7], if the regional process is stationary and under specific conditions of process support, that an optimal inverse filter solution, in the MMSE sense, for the original process may be found. The drawbacks to the proposed solution are that the filter support with few exceptions will be of infinite extent. Further a priori knowledge of the 2-D correlation function or power spectrum and possibly the use of spectral factorization techniques are required to find the filter. Physical realization of the filter would, in any event, dictate truncation of the resultant filter support region prior to implementation. Once the filter support region has been truncated, it no longer is capable of matching the original regional autocorrelation function and may not be minimum phase and therefore not stable. [Refs. 8 and 9]

The above considerations prompted the use of reduced order models for the regional processes and adaptive techniques for fitting the reduced order model. The support of the filter coefficients of the reduced order model is causal and has finite support as shown in Figure 2-3.

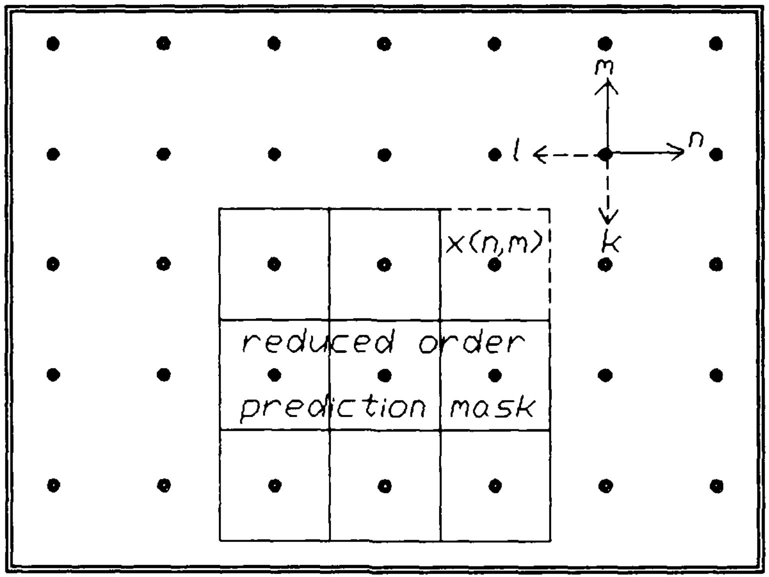

Figure 2-3. Reduced Order Model Filter Support

The resultant reduced order model is

$$x(n,m) = \sum_{\ell} \sum_{k} h(\ell,k)\; x(n-\ell,m-k) + v(n,m) \qquad (\ell,k) \in \alpha \tag{2-2}$$

where v(m,n) is a noise source, not necessarily white, that would be required to exactly reproduce the original regional process.

The driving noise process may also be viewed as the error residual in the process produced by the inverse of the reduced order model operating on the original process as shown below.

$$e(n,m) = x(n,m) - \sum_{\ell} \sum_{k} h(\ell,k)\; x(n-\ell,m-k) \qquad (\ell,k) \in \alpha \tag{2-3}$$

This representation is the prediction error filter (PEF) form [Ref. 9]. In practice if the support of the filter coefficients in the previous model is causal, and sufficiently large, the process complex spectral density function is analytic in the neighborhood of the unit bicircle, and a minimum error variance fitting technique is employed to fit the PEF to the process; then the spectral density function can be closely matched. In this situation, the noise term e(m,n) is very nearly white and gaussian if the original process is gaussian.

In practice the image background can only be assumed homogeneous (i.e., stationary) in a small region. Consequently the model will have to be fitted to each local region or adapted during the image processing. Figure 2-4 depicts a section of an image used for this processing. The point x(n,m) is the point at which the model of the process as presented in (2-3) is to be fitted. The prediction window represents the nonzero support of the summation term in (2-3) which will be referred to as the predictor. The prediction error filter is defined with a finite rectangular

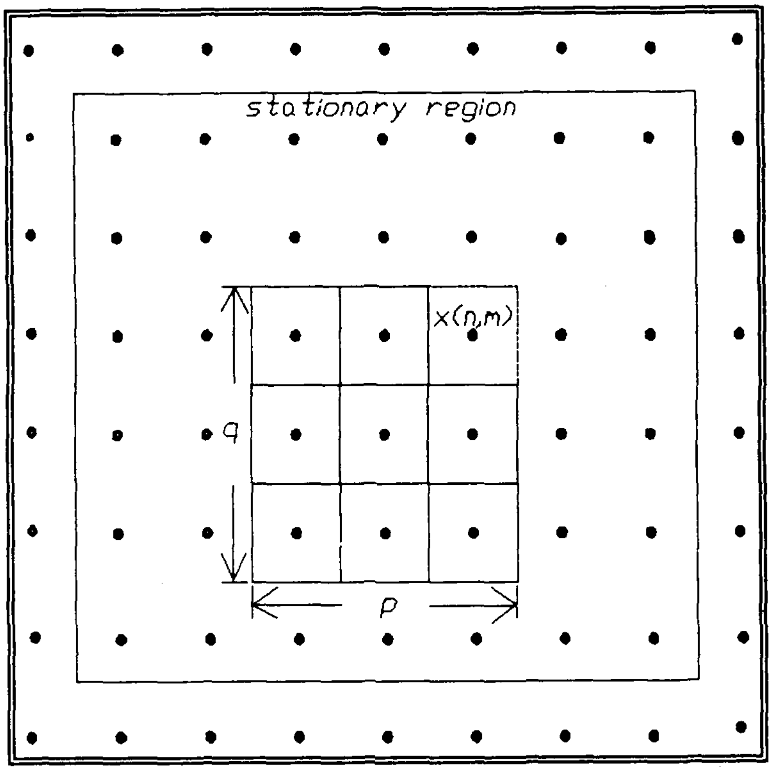

Figure 2-4. Reduced Order Model Fitting

support of size PxQ and for convenience of notation the coefficients are assigned as follows

$$a(0,0) = 1 \qquad (\ell,k) = (0,0)$$
$$a(\ell,k) = -h(\ell,k) \quad 0 \le \ell \le P-1, \; 0 \le k \le Q-1 \qquad (2-4)$$

The points underlying the prediction error filter mask may be concatenated into a column vector

$$\underline{\mathbf{x}} = \begin{bmatrix} x(n,m) \\ x(n-1,m) \\ \cdot \\ \cdot \\ x(n-P+1,m) \\ x(n,m-1) \\ x(n-1,m-1) \\ \cdot \\ \cdot \\ \cdot \\ x(n,m-Q+1) \\ x(n-1,m-Q+1) \\ \cdot \\ \cdot \\ x(n-P+1,m-Q+1) \end{bmatrix}$$

(2-5)

The point ordering establishes the past (i.e., causality) with respect to the point x(n,m). From this vector the correlation matrix can be defined as

$$\mathbf{R}_x = E\{\underline{\mathbf{x}}\ \underline{\mathbf{x}}^T\}$$

(2-6)

Since the region is assumed stationary, the correlation matrix is block Toeplitz. The optimal prediction coefficients are determined by solving the Normal Equations for linear prediction as shown below:

$$\mathbf{R}_x\underline{\mathbf{a}} = \sigma_e^2\underline{\mathbf{i}}$$

(2-7)

where $\underline{i}$ is a unit vector that is equal to one in the first row and zero elsewhere. The term σ_e^2 represents the minimum error variance, in the MMSE sense, of the white residual error of the PEF using the optimal coefficients. In this thesis the

2-D Least Mean Square (LMS) algorithm is used to obtain an approximate solution to (2-7) in a local area.

Under the assumption of local stationarity it is also possible to produce a set of prediction error filter coefficients for the region directly from the data. The method employed to produce that solution is to construct a family of equations of the form of (2-3) which represent the realization of the prediction error filter at each point in the region. These equations are then solved for the filter coefficients subject to the constraint that the sum of their error variances is minimized. The quantity to be minimized is:

$$J(\underline{\mathbf{x}},\underline{\mathbf{a}}) = E\{\sum_{\substack{n \\ (n,m) \in R}} \sum_{m} [x(n,m) - \hat{x}(n,m)]^2\} \tag{2-8}$$

where $\hat{x}(n,m)$ is the prediction estimate (2-9) for the point at position (n,m) in region R, which has been previously defined as a region in which the process may be considered stationary, given by

$$\hat{x}(n,m) = \sum_{\substack{\ell \\ (\ell,k) \in \alpha \\ (\ell,k) \neq (0,0)}} \sum_{k} h(\ell,k)\; x(n-\ell,m-k) \tag{2-9}$$

This procedure, known as "least squares," has been employed in many applications. In particular Ref. 10 has specific bearing on this thesis. The recursive form of the least squares (RLS) technique has been used in the thesis.

The final result of either the LMS or the RLS method is a "whi‿ening" of the background process. The details of the actual processing will be covered in the remainder of the thesis.

C. SIGNIFICANCE TESTING

Under hypothesis testing a threshold is established which delineates the boundary between two hypothesises H_0 and H_1. This can be seen to be a two sided testing procedure. The threshold is established by choosing the accceptable probability of rejecting H_0 when H_0 is true, P(type I error) and minimizing the probability of choosing H_0 when H_1 is true, P(type II error). This method presupposes known distributions for the two hypotheses. It has been stated previously that a robust target detection system should minimize the specifications on the target process. Hypothesis testing requires too much to be known about the target.

Significance testing is a one sided testing process with only one hypothesis H_0, refered to as the "null hypothesis." The procedure is to completely specify the probability distribution of H_0, then compute a test statistic from the event to be tested against H_0. The test statistic is a function of the event and is based solely on the event. The decision to accept or reject H_0 after the statistic is compared to H_0 is subjective, in that the establishment of boundary between the two decisions is at the descretion of the designer. Using the test statistic and the hypothesized probability distribution of H_0 a region of values for the test statistic is determined where H_0 will be rejected. This determines the probability of a type I error, which is known as the significance level of the test. The significance level may be based on desires to minimize the false alarm rate (time between type I errors), maintain a constant false alarm rate, etc. [Ref. 6]

The significance testing used for target detection in this thesis involves a hypothesized gaussian distributed white noise background (H_0). This noise is ideally the result of decorrelating the background as previously mentioned. The specifics of the testing will be covered in Chapter V.

III. LMS FOR TWO-DIMENSIONAL PREDICTION

A. THEORY

In the previous chapter the use of a PEF with a reduced order predictor model was proposed as a method to decorrelate the background process. This, however, was contingent on choosing the predictor coefficients such that the error variance at the output of the PEF was minimized, see Figure 3-1.

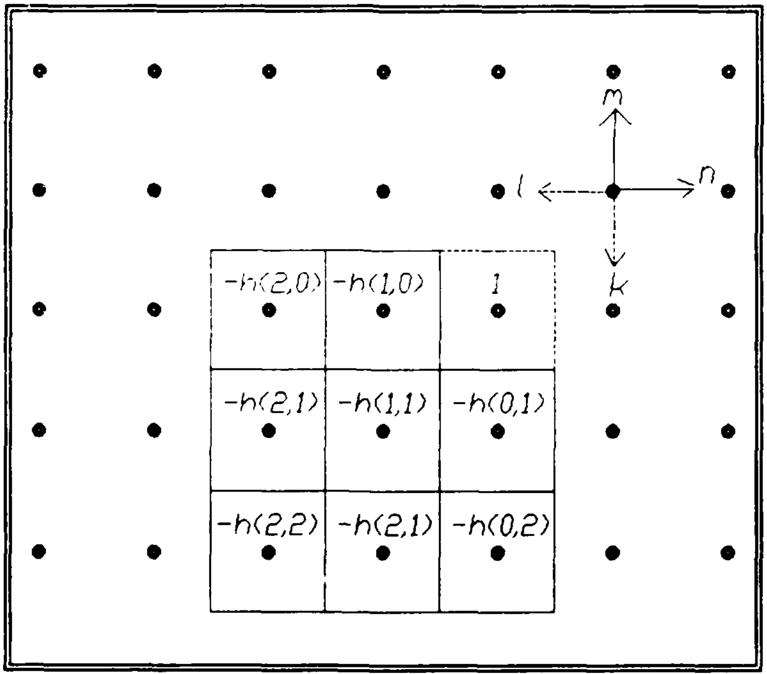

Figure 3-1. Prediction Error Filter

This chapter extends the traditional LMS algorithm to the problem of two-dimensional PEF. In this form the coefficients of the predictor are adaptively changed to minimize the mean squared error. [Refs. 11 and 12] The criterion to be minimized is

$$\xi = E[e(n,m;i)^2] \tag{3-1}$$

The error term e(n,m;i), for the predictor mask of size PxQ with first quadrant support shown in Figure 3-1, is defined as

$$e(n,m;i) = x(n,m) - \sum_{\substack{\ell=0 \\ (\ell,k) \neq (0,0)}}^{P-1} \sum_{k=0}^{Q-1} h(\ell,k;i)\ x(n-\ell,m-k) \tag{3-2}$$

where the index i represents the i^{th} update of the filter. If the error is squared the criterion becomes

$$\xi = E\{x(n,m)^2\} - 2E\{x(n,m) \sum_{\ell=0}^{P-1} \sum_{k=0}^{Q-1} h(\ell,k;i)\ x(n-\ell,m-k)\}$$
$$+E\{ \sum_{\substack{g=0 \\ (g,h) \neq (0,0) \\ (\ell,k) \neq (0,0)}}^{P-1} \sum_{h=0}^{Q-1} \sum_{\ell=0}^{P-1} \sum_{k=0}^{Q-1} h(g,h;i)\ h(\ell,k;i)\ x(n-g,m-h)\ x(n-\ell,m-k)\} \tag{3-3}$$

The signal x(n,m) is assumed to be stationary and ergodic. It will be shown later that the stationarity assumption can be relaxed and still produce adequate results. The result of

(3-3) is a quadratic error surface that has a unique minimum. Differentiating (3-3) with respect to the predictor coefficients yields

$$\frac{\partial \xi}{\partial h(\ell,k;i)} = \nabla(i) = -2E\{[x(n,m) - \sum_{g=0}^{P-1} \sum_{h=0}^{Q-1} h(g,h;i)\, x(n-g,m-h)] \cdot \mathbf{x(n-\ell,m-k)}\}$$

$$0 \le \ell \le P-1$$
$$0 \le k \le Q-1$$
$$(\ell,k) \ne (0,0)$$

(3-4)

This is the gradient of the error surface at update i mapped into the predictor coefficient space. The term **x(n-ℓ,m-k)** represents the array of image points under the prediction mask; the gradient ∇(i) also is in array form. At the minimum point on the surface the gradient vanishes. The result of minimizing (3-4) and substituting into (3-3) produces the following expression for the minimum mean square error (MMSE):

$$\xi_{min} = E\{x(n,m)^2\} - E\{x(n,m) \sum_{\ell=0}^{P-1} \sum_{k=0}^{Q-1} h(\ell,k;i)\, x(n-\ell,m-k)\}$$

(3-5)

B. THE STEEPEST DESCENT ALGORITHM (SDA)

Central to the theme of the LMS algorithm is the method of steepest descent. This exploits the quadratic character of the error surface by iterative adjustment of the filter coefficients. The coefficients are adjusted by an amount proportional and opposite in direction to the gradient at the current location on the error surface, that is

$$h(\ell,k;i+1) = h(\ell,k;i) + \mu(-\nabla(i)) \qquad \begin{array}{l} 0 \le \ell \le P-1 \\ 0 \le k \le Q-1 \\ (\ell,k) \ne (0,0) \end{array} \tag{3-6}$$

Here, as in the one dimensional LMS [Ref. 11], the gradient is approximated by removing the expectation operator from (3-4) and recognizing the bracketed term as the prediction error, e(n,m;i), at update i. The form of the coefficient update (3-6) now becomes

$$h(\ell,k;i+1) = h(\ell,k;i) + 2\mu\{e(n,m;i)\ x(n-\ell,m-k)\} \qquad \begin{array}{l} 0 \le \ell \le P-1 \\ 0 \le k \le Q-1 \\ (\ell,k) \ne (0,0) \end{array} \tag{3-7}$$

The term μ is a scalar known as the step size and controls the rate of adaption [Ref. 11].

C. ALGORITHM ANALYSIS

In order to implement the LMS algorithm it necessary to use (3-2) and (3-7), and to choose an appropriate value for μ and an initial condition on the coefficient array.

The LMS algorithm then attempts to track the error surface, under the assumption of stationarity, by adjusting the filter coefficients. The result is an error surface that is inherently changing even if the original image is stationary.

In any analysis of LMS there is the problem of nonstationarity of the data over the entire image frame that needs to be considered. This problem has been dealt with from a variety of view points. In addition image processing has inherent problems caused by the overlapping of data blocks which was assumed. In the original derivation of the one dimensional LMS algorithm, independence of successive input data blocks is frequently assumed. This is not necessarily true for image data. This problem has been addressed in the literature under the assumption of specific dependencies. These range from independent input data to input data possessing considerable correlation. [Refs. 13-17] An expansion on the analysis is beyond the scope of this thesis. Experimental results indicate that Widrow's method of dealing with nonstationarity [Ref. 11], as adapted by Bitmead [Ref. 18] and Hsia [Ref. 19] and termed "The Normalized LMS (NLMS)," has some general applicability in the processing of two-dimensional images. As originally presented the algorithm was based on the assumption of gaussian independent identically distributed (iid) data. In order to present an overview of this method, as adapted to two-dimensional processes, define

$$\underline{x}(i) = \begin{bmatrix} x(n-1,m) \\ x(n-2,m) \\ \vdots \\ x(n-P+1,m) \\ \vdots \\ x(n,m-Q+1) \\ x(n-1,m-Q+1) \\ \vdots \\ x(n-P+1,m-Q+1) \end{bmatrix} \tag{3-8}$$

This vector consists of the inputs under the PxQ prediction mask at update i. Further define

$$\underline{h}(i) = \begin{bmatrix} h(1,0) \\ h(2,0) \\ \vdots \\ h(P-1,0) \\ \vdots \\ h(0,0-1) \\ h(1,Q-1) \\ \vdots \\ h(P-1,Q-1) \end{bmatrix} \tag{3-9}$$

as the prediction coefficients at update i concatenated into a column vector. Using the vector form of (3-7) it was proposed by Hadhound and Thomas [Ref. 12] that the actual coefficient adaption process in a image could be represented as

$$\underline{h}^{\circ}(i+1) = \underline{h}^{\circ}(i) + \underline{\gamma}(i)$$

(3-10)

The term $\underline{\gamma}(i)$ is a column vector of random increments to the ideal time varying filter which would track the input data. Subtracting (3-10) from (3-7) in vector form one can define

$$\underline{V}(i) = \underline{h}(i) - \underline{h}^{\circ}(i)$$

(3-11)

which represents the coefficient deviation from the ideal coefficients. The vector $\underline{V}(i)$ is called the coefficient error vector. The coefficient error update can be written as follows

$$\underline{V}(i+1) = (I-2\mu\underline{x}(i)\underline{x}(i)^{T})\underline{V}(i) + 2\mu\underline{x}(i)e^{\circ}(i)-\underline{\gamma}(i)$$

(3-12)

where $e^{\circ}(i)$ is the error resulting from prediction with the optimum time varying coefficients $\underline{h}^{\circ}(i)$. The coefficient vector $\underline{h}^{\circ}(i)$ represents the optimal coefficients at update i as determined by the solution of the normal equations. The ultimate desire is for the coefficient error term to approach zero as i becomes large. If the input data is independent and stationary, this can be accomplished [Ref. 11] with a step size $0<\mu<{}^{1}/_{\lambda_{max}}$, where λ_{max} is the largest eigenvalue of $\mathbf{R}$. However in the current form (3-12) it can be seen that this would not produce zero coefficient error as time progressed. The result is a contribution to the coefficient deviation due to the approximation of the gradient (the first term of (3-12)) and a deviation due to the lag in tracking the ideal coefficient changes (the last terms in (3-12)) [Refs. 15 and 16].

In the "Normalized LMS (NLMS)" algorithm the step size is defined as

$$\mu = \frac{\alpha}{4\mathrm{Tr}(\mathrm{R})} \tag{3-13}$$

where Tr(**R**) may be approximated by the $\|\underline{x}(i)\|^2$. The form given in (3-12) after substituting (3-13) has both heuristic and practical appeal. Both terms containing the input data under the mask have been normalized. This decreases the explicit dependence of the coefficient error on the change of data input power. The term α has been addressed in [Ref. 14] for the iid case and related to the ratio of the variance of the step change in (3-10) and the MMSE. This is of interest from a theoretical standpoint; however, in practice these factors are rarely available. In practice it has been observed that choosing α to satisfy $0<\alpha<2$ for the case of independent data works well with the most rapid convergence of the MSE at $\alpha=1$. In Ref. 14 an analysis was performed on correlated data for a two input predictor. The results extended to the "Normalized LMS" algorithm would indicate that the greater the correlation of the input data the smaller should be the α term (typically $\alpha<1$). After substitution of the step size (3-13) the coefficient update in vector form becomes

$$\underline{h}(i+1) = \underline{h}(i) + \frac{\alpha e(n,m;i)\underline{x}(i)}{2Tr(R)} \tag{3-14}$$

The coefficient error results in excessive MSE. A measure of this is the misadjustment ratio M defined as

$$M = \frac{\epsilon(\infty) - \epsilon_{min}}{\epsilon_{min}} \tag{3-15}$$

where $\epsilon(\infty)$ is the MSE at steady state and ϵ_{min} is the theoretical MMSE [Ref. 14].

Figure 3-2 shows a plot of the misadjustment ratio versus various values of the α term in the step size. The results presented are the average of simulations on synthetic images. The image process is modeled by a 2x2 quarter plane separable autoregressive filter.

$$x(n,m) = 0.5\ x(n-1,m) - 0.4\ x(n,m-1) + 0.8\ x(n-1,m-1) + w(n,m) \tag{3-16}$$

The results are for a range of σ_w^2 from 0.6 to 4.0. It can be seen that this form of the step size produces a unique minimum M for a given process model. Considering (3-12) with Figure 3-2, it can also be seen that as α becomes small, the excess

MSE increases due to coefficient lag. Further, as α increases, so does the excess MSE due to gradient error.

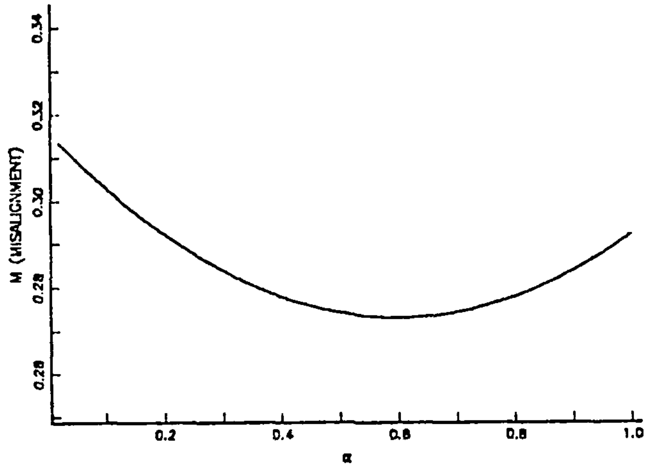

Figure 3-2. Misadjustment Versus α

An example of the predictor coefficient deviation is shown in Figure 3-3. The image frame processed with the NLMS PEF was synthesized with (3-16) and σ_w^2=0.64.

The initial coefficients were set by a least squares technique discussed in the implementation section. It can be seen for the processed row that the coefficients, while not the same as those for the original process, are stable and tracking. The row processed for Figure 3-3 is shown in Figure 3-4.

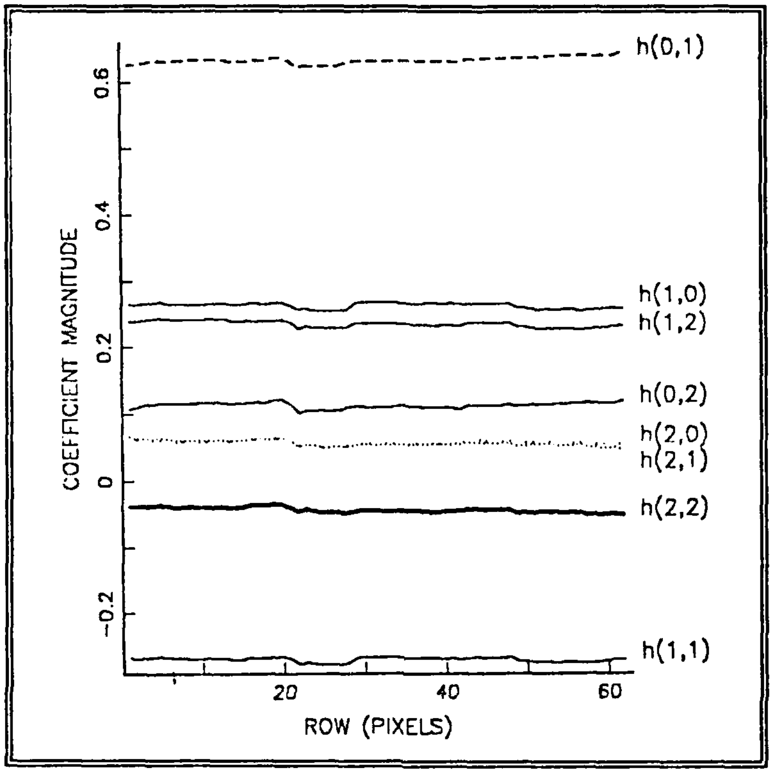

Figure 3-3. Predictor Coefficient Plot

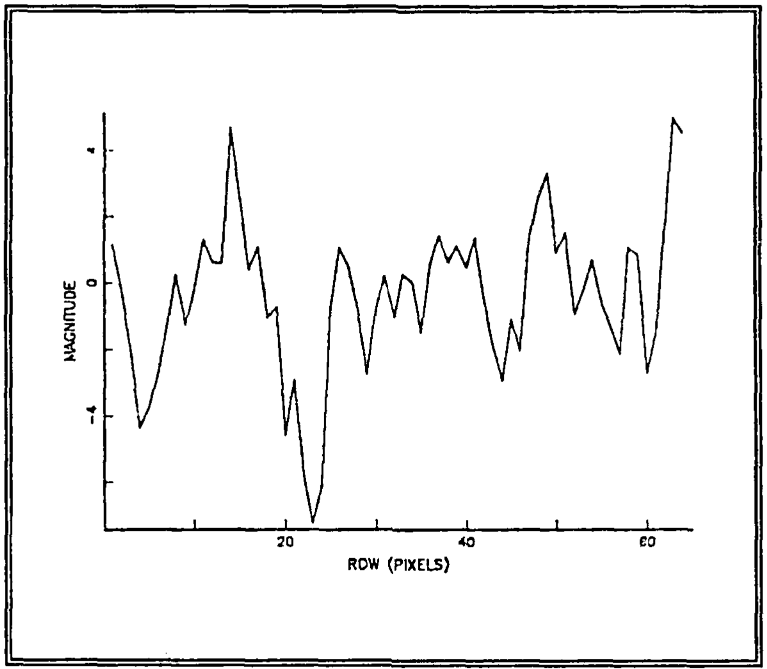

Figure 3-4. *Row of the Background* Process

The desired result is the decorrelation of the image process (i.e., a nearly white noise error residual); this can be seen in Figure 3-5.

Figure 3-6 contains a plot of the normalized histogram of the error residual output of the NLMS Prediction Error Filter.

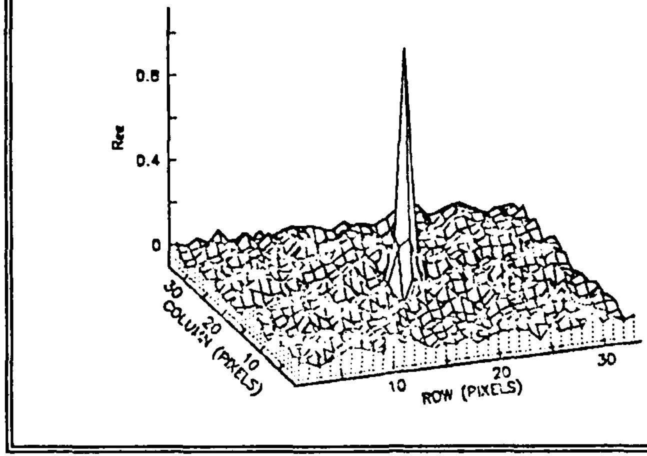

Figure 3-5. Error Residue Correlation Function

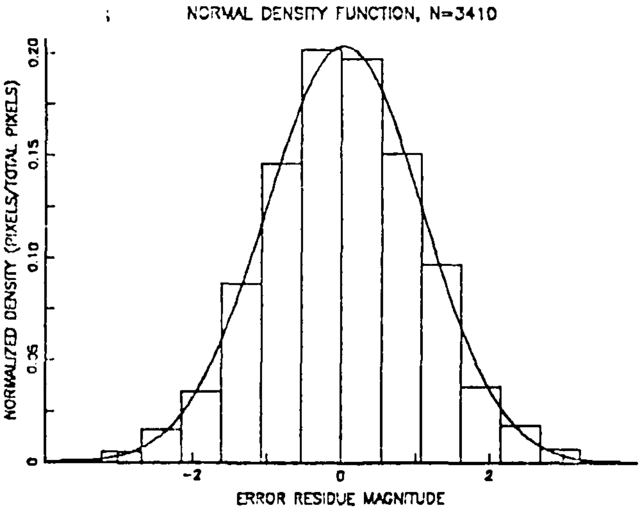

Figure 3-6. Error Residue Normalized Histogram Plot with Normal Density Function Superimposed

It can be seen that the error residual distribution has a gaussian appearance as expected with a gaussian driving process.

D. LMS PEF IMPLEMENTATION

One of the deficiencies in the LMS algorithm is the speed of convergence. The detection algorithm discussed in a later chapter depends on the ability of the PEF to decorrelate the background process.

Another problem that is inherent in the processing of the image is that of directionality. In order to process the image a scan pattern and a method of handling boundary conditions must be chosen. Once these have been established, discontinuities and directional dependencies have been introduced into the process. Although directional dependence cannot be overcome, the effect of discontinuities and settling time can be decreased.

The processing of images for this thesis was done using a PEF with a 3x3 rectangular support and a row raster scan pattern. Initial conditions were divided into two parts. The filter coefficients at the beginning of each frame were initialized with coefficients derived from a least squares solution for a small area (9x9) pixels (see Figure 3-7) at the beginning of the frame. The initial coefficients could have been derived from other methods. However, the method used

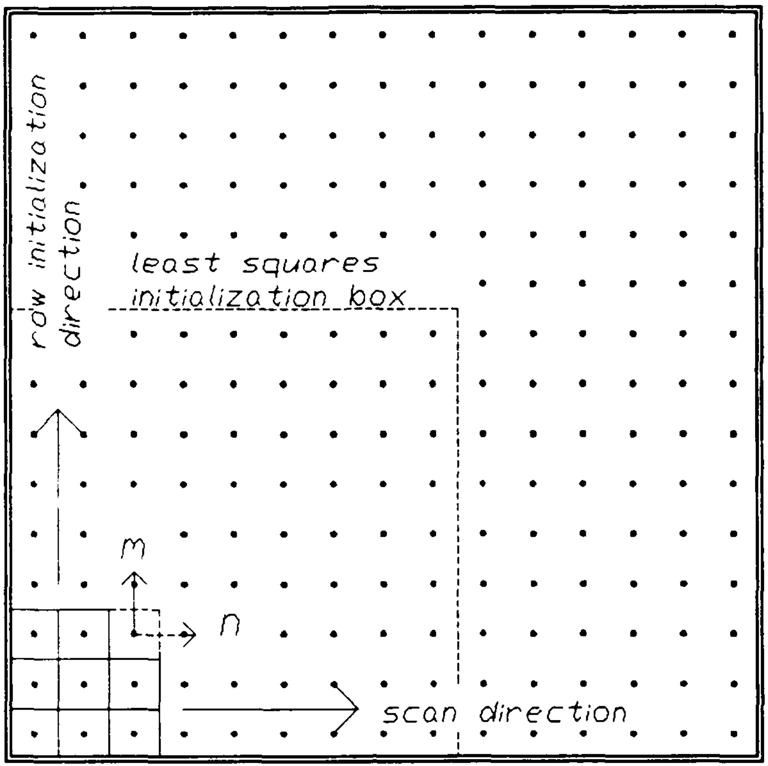
row initialization direction
least squares initialization box
m
n
scan direction

Figure 3-7. Process Scan Diagram With Initialization Window

should have a misadjustment less than or equal to that anticipated for the LMS algorithm. Initial coefficients for successive rows were steepest descent algorithm updates of the previous row initial coefficients.

The remainder of the frame processing follows from the solution of (3-2) and the coefficient update computed using (3-7). The online computation of an estimate of the misadjustment ratio can be used as a measure of the algorithm efficiency. Anomalous jumps in the misadjustment can be used to signal that the coefficients should be reinitialized using the least squares method.

IV. RECURSIVE LEAST SQUARES FOR TWO-DIMENSIONAL PREDICTION

A. THEORY

The least squares algorithm is a method used in the estimation of model parameters. [Refs. 6, 22, and 23] The objective is the estimation of the model parameters that will fit the model to the observed data. The criterion for determining the goodness of fit is the minimum sum of the squared error between the model output and all observations. An extension of this method is the Recursive Least Squares (RLS) algorithm. This method is identical to the least squares algorithm but provides for updating the estimate of the model parameters as more data becomes available. In Chapter II the concept of fitting a reduced order model (prediction model) to an image was addressed. This model will be used in the RLS algorithm and has the form

$$x(n,m) = \sum_{\ell}\sum_{k} h(\ell,k)\; x(n-\ell,m-k) + e(n,m) \qquad \begin{matrix} (\ell,k) \in \alpha \\ (\ell,k) \neq (0,0) \end{matrix} \tag{4-1}$$

The term $x(n,m)$ is the observation and the terms $x(n-\ell,m-k)$ are the past data. The terms $h(\ell,k)$ are the unknown parameters that must be determined in order to fit the model to the observation. The term e(n,m) is the error between the model and the observed image data at position (n,m). The least squares criterion is as follows

$$J(h,x;N,M) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} e^2(h,x;n,m)$$

(4-2)

where J(h,x;N,M) is the cost function to be minimized by an appropriate choice of $h(\ell,k;N,M)$. The error term in (4-2) is defined as

$$e(h,x;n,m) = x(n,m) - \hat{x}(n,m)$$

(4-3)

The term $\hat{x}(n,m)$ is the prediction of the point x(n,m) defined as

$$\hat{x}(n,m) = \sum_{\substack{\ell=0 \\ (\ell,k)\neq(0,0)}}^{P-1} \sum_{k=0}^{Q-1} h(\ell,k;N,M)\ x(n-\ell,m-k)$$

(4-4)

where the support of the predictor model is PxQ. The second indices (N,M) in the term $h(\ell,k;N,M)$ are included to indicate that $h(\ell,k;N,M)$ is held constant for a block of data $0\leq n\leq N-1$ and $0\leq m\leq M-1$. Thus it is the sum of the errors with this choice for the coefficients that is minimized.

The criterion as defined above in conjunction with the prediction model can be recast in the form of a PEF. Figure 4-1 shows a 3x3 rectangular PEF as defined by (4-5).

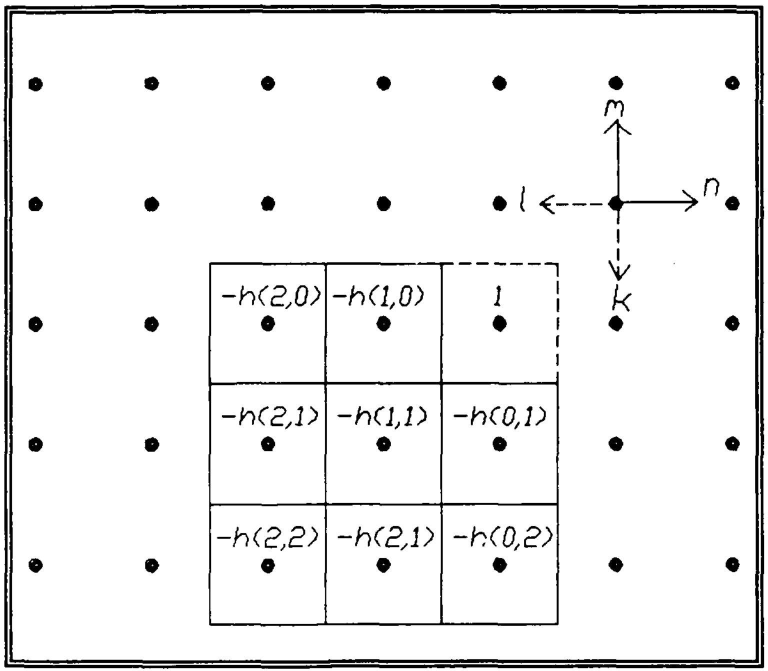

Figure 4-1. Prediction Error Filter

$$e(n,m) = x(n,m) - \sum_{\substack{\ell=0 \\ (\ell,k)\neq(0,0)}}^{P-1} \sum_{k=0}^{Q-1} h(\ell,k;N,M)\, x(n-\ell,m-k) \tag{4-5}$$

The output of the PEF is the error term to be minimized as defined by the cost function. In order to simplify the notation the h and x variables have been dropped from the error term; however, the dependency is understood. As proposed in Chapter II, if the error term is minimized and the original process was a gaussian WNDR, then the error will be white and gaussian distributed.

At this point it is advantageous to define the previous expressions in vector form. Although the ultimate objective and results remain the same, directionality is inherent in the form to follow. The development to follow will be for a raster scan by row, but it may be generalized for any scan pattern for which recursive computability is assured. The mapping function for a row scan (see Figure 4-1) is

$$i = n + Nm \quad 0 \leq n \leq N-1, \; 0 \leq n \leq N-1 \tag{4-6}$$

where N represents the row length and M represents the column length of the data field defined on a rectangular lattice. By using this mapping function to replace the indices (n,m) and concatenating terms by rows into a column vector the following terms can be redefined.

$$\underline{x}(i) = \begin{bmatrix} x(n-1,m) \\ x(n-2,m) \\ \cdot \\ \cdot \\ x(n-P+1,m) \\ \cdot \\ \cdot \\ \cdot \\ x(n,m-Q+1) \\ x(n-1,m-Q+1) \\ \cdot \\ \cdot \\ x(n-P+1,m-Q+1) \end{bmatrix} \tag{4-7}$$

and

$$\underline{h}(j) = \begin{bmatrix} h(1,0) \\ h(2,0) \\ \cdot \\ \cdot \\ h(P-1,0) \\ \cdot \\ \cdot \\ \cdot \\ h(0,Q-1) \\ h(1,Q-1) \\ \cdot \\ \cdot \\ h(P-1,Q-1) \end{bmatrix}$$

(4-8)

The vector $\underline{x}(i)$ is thus comprised of the points under the prediction mask at update $i=n+Nm$. The vector $\underline{h}(j)$ is comprised of the prediction filter coefficients based on all data for $0 \le i \le j-1$. It follows from this mapping that

$$x(i) = x(n,m) \text{ and } \hat{x}(i) = \hat{x}(n,m)$$

(4-9)

and

$$\hat{x}(i) = \underline{h}^T(j)\ \underline{x}(i) \qquad 0 \le i \le j-1$$

(4-10)

The cost function is then expressed as

$$J(h,x;j) = \sum_{i=0}^{j-1} \lambda^{j-(i+1)} e(i)^2 = \sum_{i=0}^{j-1} \lambda^{j-(i+1)} (x(i)-\underline{h}^T(j)\underline{x}(i))^2$$

(4-11)

Where the term $\lambda^{j-(i+1)}$ is added to give the cost function a fading memory [Ref. 23] and thus a greater ability to adapt to nonstationarities in the image. In a stationary image the exponential weighting factor would be set to unity. From this point on, the development of the RLS algorithm will parallel Haykin's one dimensional approach in Refs. 22 and 23.

The cost function as presented is quadratic in the terms of $\underline{h}(j)$ and therefore possesses a unique minimum. To solve for the optimal coefficients $\underline{h}(j)$ in terms of the cost function, (4-11) is differentiated with respect to the filter coefficients.

$$\frac{\partial J(h,x;j)}{\partial \underline{h}(j)} = -2 \sum_{i=0}^{j-1} \lambda^{j-(i+1)} \underline{x}(i) - \sum_{i=0}^{j-1} \lambda^{j-(i+1)} \underline{h}(j)\underline{x}(i) \tag{4-12}$$

Now as in Ref. 23 the following definitions will be made. The deterministic correlation matrix is defined as

$$\Phi(j) = \sum_{i=0}^{j-1} \lambda^{j-(i+1)} \underline{x}(i)\underline{x}(i)^{T} \tag{4-13}$$

In the same vein the deterministic cross correlation vector (i.e., the cross correlation of the point to be predicted with the data vector) is defined as

$$\underline{\theta}(j) = \sum_{i=0}^{j-1} \lambda^{j-(i+1)} x(i)\underline{x}(i) \tag{4-14}$$

Using the last two definitions, setting the derivative of the cost function (4-12) equal to zero and rearranging yields

$$\Phi(j)\ h(j) = \underline{\theta}(j)$$

(4-15)

This form has been termed in Refs. 22 and 23 as the deterministic normal equations. The solution of (4-15) produces the optimal coefficients $\underline{h}(j)$ in the least squares sense.

$$\underline{h}(j) = \Phi^{-1}(j)\ \underline{\theta}(j)$$

(4-16)

The objective now is to solve for $\underline{h}(j)$ recursively. This can be accomplished by forming the right hand side of (4-16) recursively. [Ref. 22] From (4-13) the correlation matrix can be expressed in the recursive form

$$\Phi(j) = \lambda\Phi(j-1) + \underline{x}(j)\underline{x}(j)^{T}$$

(4-17)

Further, from (4-14), the cross correlation vector may be formed recursively as

$$\underline{\theta}(j) = \lambda\ \underline{\theta}(j-1) + x(j)\underline{x}(j)$$

(4-18)

In forming the inverse of the correlation matrix the matrix inversion lemma [Ref. 22] is used. On application of the lemma, the inverse of the correlation matrix becomes

$$\Phi^{-1}(j) = \lambda^{-1}\Phi^{-1}(j-1) - \frac{\lambda^{-2}\Phi^{-1}(j-1)\ \underline{x}(j)\underline{x}(j)^{T}\Phi^{-1}(j-1)}{1 + \lambda^{-1}\underline{x}^{T}(j)\ \Phi^{-1}(j-1)\underline{x}(j)}$$

(4-19)

From (4-19) two terms are defined which will facilitate computations. These are

$$\mathbf{P}(j) = \Phi^{-1}(j) \tag{4-20}$$

and

$$\underline{K}(j) = \frac{\lambda^{-1}\ \mathbf{P}(j-1)\ \underline{x}(j)}{1 + \lambda^{-1}\underline{x}^T(j)\ \mathbf{P}(j-1)\underline{x}(j)} \tag{4-21}$$

where $\underline{K}(j)$ is called the "gain vector." The form of $\underline{K}(j)$ can be shown [Ref. 22] to reduce to

$$\underline{K}(j) = \mathbf{P}(j)\ \underline{x}(j) \tag{4-22}$$

It also follows from (4-19, 20, 21 and 22) that

$$\mathbf{P}(j) = \lambda^{-1}\ \mathbf{P}(j-1) - \lambda^{-1}\underline{K}(j)\ \underline{x}^T(j)\ \mathbf{P}(j-1) \tag{4-23}$$

The update of the filter coefficient vector $\underline{h}(j)$ can now be formed with the use of (4-16, 17, 21, and 22).

$$\underline{h}(j) = \underline{h}(j-1) + \underline{K}(j)\ [\underline{x}(j) - \underline{h}^T(j-1)\ \underline{x}(j)] \tag{4-24}$$

Using this equation the prediction error is defined as

$$e(j) = x(j) - \underline{h}^T(j-1)\underline{x}(j) \tag{4-25}$$

This has been termed the a priori prediction error. Further define the a posteriori prediction error as

$$\alpha(j) = x(j) - \underline{h}^T(j)\underline{x}(j)$$

(4-26)

It is the a prior prediction error that is used in the cost function in (4-2). Now using (4-25) in (4-24) the final form of the coefficient update equation is

$$\underline{h}(j) = \underline{h}(j-1) + \underline{K}(j)e(j)$$

(4-27)

This equation completes the necessary equations for the construction of the adaptively updated prediction error filter.

A term that will be useful in the detection algorithm (covered in the next chapter) is the minimum value of the cost function. The cost function attains the minimum value if at each step in the recursion the filter coefficients satisfy the deterministic normal equations (4-15). In this case the cost function may be written as

$$\mathbf{J}_{min}(j) = \lambda^{j-(i+1)}\, \mathbf{J}_{min}(j-1) + e(j)\alpha(j)$$

(4-28)

The weighted average of $\mathbf{J}_{min}(j)$ over the range of the iterations modified by the exponential weight factor provides an estimate of the error variance at step j. The effective memory length for averaging purposes is

$$\mathbf{MEM} = \frac{\lambda}{1-\lambda}$$

(4-29)

B. ALGORITHM ANALYSIS

The properties of the least squares method of parameter estimation have been addressed extensively in Ref. 23 and Ref. 24. In Chapter II it was assumed that the background process could be modelled as a WNDR. It was further assumed that, if our reduced order model was sufficiently large and a minimum variance fitting technique was employed, the resultant error would be white. Under these assumptions the least squares estimate of $h(\ell,k;M,N)$ has some important properties, namely that it is consistent and the best linear unbiased estimator. This has been shown for the one-dimensional least squares problem in [Refs. 23, 24, and 25] and extends readily to the two-dimensional problem. Further it has been shown [Ref. 22 and 23] for the one dimensional case that the mean square error will converge in approximately twice the filter length to the minimum mean square error. This was shown experimentally to be a good estimate for the two dimensional case as well.

In addition, if the driving source for the model is gaussian, then the least squares estimate results in the maximum likelihood estimate. In order to demonstrate this some terms will need to be defined. Assume as before that the error residual is white and gaussian. The joint probability distribution for the error residual in (4-11), without the exponential weighting factor is

$$p(e(0),e(1),\ldots e(j-1)) = \frac{1}{(2\pi\sigma_e^2)^{j/2}} \exp\left[\frac{-\sum_{i=0}^{j-1} e(i)^2}{2\sigma_e^2}\right] \tag{4-30}$$

Inserting (4-25) for the term e(i) the likelihood function can be written as

$$L_x(\underline{h}(j),\sigma_e^2) = \frac{1}{(2\pi\sigma_e^2)^{j/2}} \exp\left[-\frac{\sum_{i=0}^{j-1} (x(i) - \underline{h}^T(j)\underline{x}(i))^2}{2\sigma_e^2}\right] \tag{4-31}$$

It can be seen that maximizing the likelihood of x(i), the observations, with respect to the parameter $\underline{h}$ to be estimated is the same as the least squares criteria.

The algorithm has its faults as well as its good points. Without the exponential weighting factor the memory length is unlimited. This presents a problem in implementation on fixed word length devices. Normalization of the **P** matrix has been attempted to overcome this problem; however, this introduces a bias in the mean square error. The long memory also makes the algorithm less responsive to changes in the image as time progresses. This was the reason for the inclusion of the exponential weighting factor in the recursions. There is a drawback to the use of the weighting factor however. The smaller the factor (i.e., shorter the memory), the greater the error variance. Experimental results with synthetic images have shown this to be true. At a memory length of less than approximately ten pixels the results were indistinguishable from those of LMS.

C. RLS PEF IMPLEMENTATION

The following is the complete RLS algorithm. Note that the initialization given here is the traditional one. The actual initialization techniques employed in this thesis are covered following the algorithm listing. The initial conditions on **P**, $\underline{h}$, and $\mathbf{J}_{min}$ are

$$\mathbf{P}(0) = \delta^{-1}\mathbf{I}$$

$$\underline{h}(0) = \underline{0}$$

$$\mathbf{J}(0) = 0$$

(4-32)

where δ is a small number (approx. <<1). The exponential weight factor is set at the desired value in the range $0 \leq \lambda \leq 1$. The update recursion for $i \geq 0$ is then

$$\underline{K}(i) = \frac{\lambda^{-1}\ \mathbf{P}(i-1)\underline{x}(i)}{1 + \lambda^{-1}\underline{x}^{T}(i)\mathbf{P}(i-1)\underline{x}(i)}$$

(4-33)

where $\underline{K}(i)$ is the gain vector. The inverse correlation matrix **P** is updated by

$$\mathbf{P}(i) = \lambda^{-1}\mathbf{P}(i-1) - \lambda^{-1}\underline{K}(i)\underline{x}^{T}(i)\mathbf{P}(i-1)$$

(4-34)

The a priori prediction error is

$$e(i) = x(i) - \underline{h}^T(i-1)\underline{x}(i) \tag{4-35}$$

The filter coefficients are updated as follows

$$\underline{h}(j) = \underline{h}(j-1) + \underline{K}(j)e(j) \tag{4-36}$$

The posteriori prediction error update is

$$\alpha(i) = x(i) - \underline{h}^T(j)\underline{x}(i) \tag{4-37}$$

The final item to be computed is J_{min}

$$J_{min}(i) = \lambda J_{min}(i-1) + e(i)\alpha(i) \tag{4-38}$$

It is from this last term that an estimate is formed of the error variance. This estimate is

$$\sigma_e^2 = J_{min}(i)/(n-8) \tag{4-39}$$

where n is the number of points processed and the n-8 term is to account for the loss of 8 degrees of freedom due to estimating the filter coefficients from the data set [Ref. 24].

The speed of convergence has been previously mentioned. While a convergence rate of twice the filter length may provide an acceptable rate of convergence for continuous data sets it presents a problem in image processing. The problem arises in three areas:

1) at the initial start up on each frame,

2) at the beginning of each row or column in a row or column raster scan, and

3) where a large change in background statistics occurs.

In order to reduce the effects of the problem areas the least square algorithm was implemented in the form of a secondary processor. Logic flags were established in the RLS processing to signal the occurrence of one of the problems. Once one of the flags is raised the RLS algorithm is re-initialized with the least squares algorithm. The least squares algorithm is applied to a 9x9 rectangular block of data centered on the flagged point in Figure 4-2. The algorithm produces an estimate of the filter coefficients, the **P** matrix, and $\mathbf{J}_{min}$ which are passed to the main RLS processing routine.

It can be seen that this procedure can introduce a considerable burden in the processing. A suitable alternative is to eliminate the row initialization flag while still retaining the frame initialization flag. The initial estimate of **P**, $\underline{h}(j)$, and $\mathbf{J}_{min}$ for each row or column could be obtained from the initial value of the previous row or column. The update is repeated for successive rows or columns in the same manner.

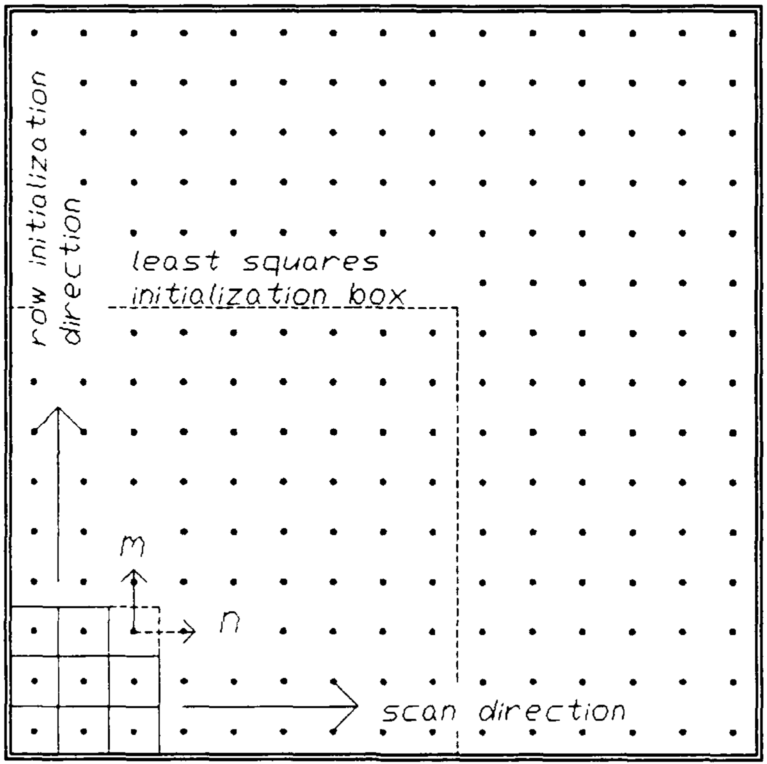
row initialization direction
least squares initialization box
m
n
scan direction

Figure 4-2. RLS PEF Initialization

An example of the experimental results follows using the two-dimensional RLS algorithm. The image processed was synthetically produced using a 2x2 autoregression mask. The actual process model was separable and defined as

$$x(n,m) = 0.5x(n-1,m) - 0.4x(n,m-1) + 0.8x(n-1,m-1) + v(n,m) \tag{4-40}$$

with the variance of the source σ_v^2=2.0 and output σ_x^2=14.1. The overall mean of the image was removed prior to processing. The correlation function of the error residual resulting from processing the entire image is shown in Figure 4-3.

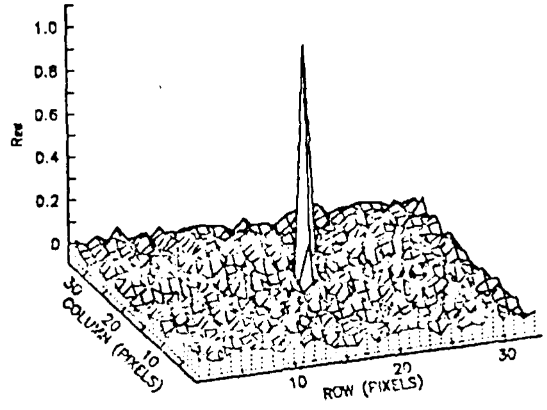

Figure 4-3. Error Residual Correlation Function

Figure 4-4 is a histogram plot of the error residual with a gaussian distribution superimposed for comparison.

Figure 4-5 is a plot of the filter coefficients resulting from a RLS row raster scan of the same row as in Figure 4-3.

This concludes the development and implementation of the two dimensional RLS algorithm. The RLS algorithm was found to be superior to the LMS algorithm in all areas except in the area of computations required. The performance will prove to be an overriding consideration in the target detection problem.

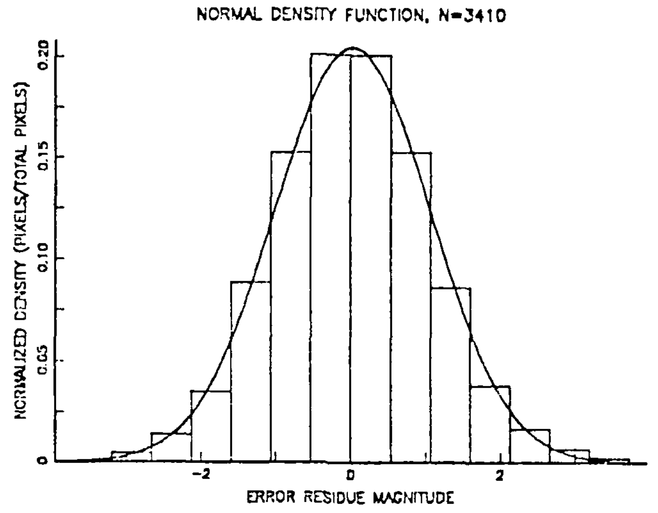

Figure 4-4. Histogram of Error Residue

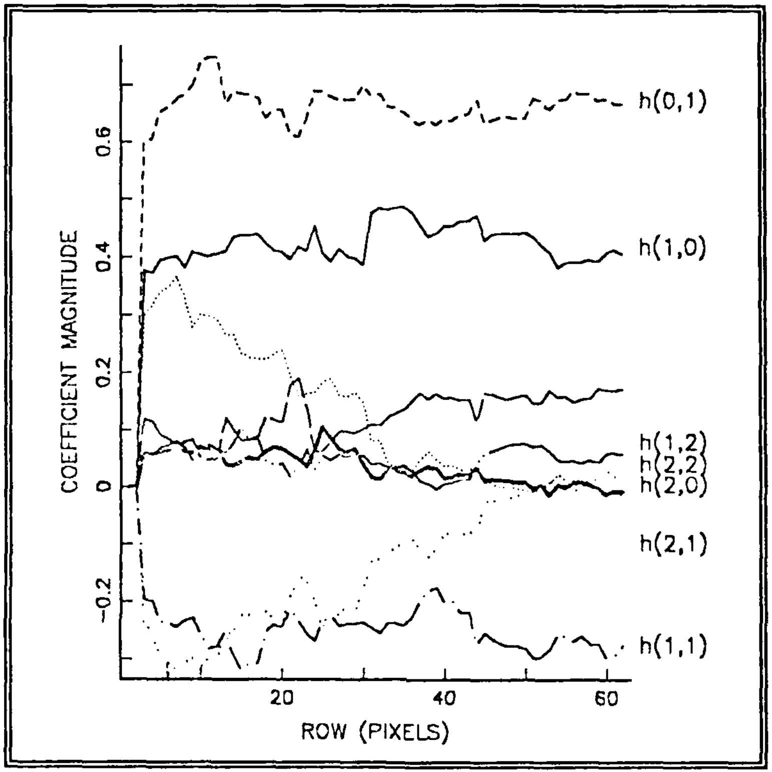

Figure 4-5. Predictor Coefficient Plot

V. TARGET DETECTION

A. GENERAL

The previous three chapters have dealt with the theory and algorithms for modelling the background process. It was shown that the result of inverse filtering with an adaptive PEF is a nearly white error residual process. It is this error residual that is of particular interest in the detection algorithm. The residual represents the unpredictable portion of the image process. The question that arises is the effect of the resultant decorrelation on point targets present in the image.

When a target is present in an image, it replaces the background pixel at that location. Therefore the target is opaque as far as the image process is concerned. Further the target, if present, is deterministic and its intensity remains constant over time. The probability density of the image versus the target may be viewed as in Figure 5-1. In this figure the intensity of the target is a fixed value T and its PDF is represented as an impulse; the mean of the PDF for the background is specified by m. It can be seen that only the magnitude of the difference between the image mean and the

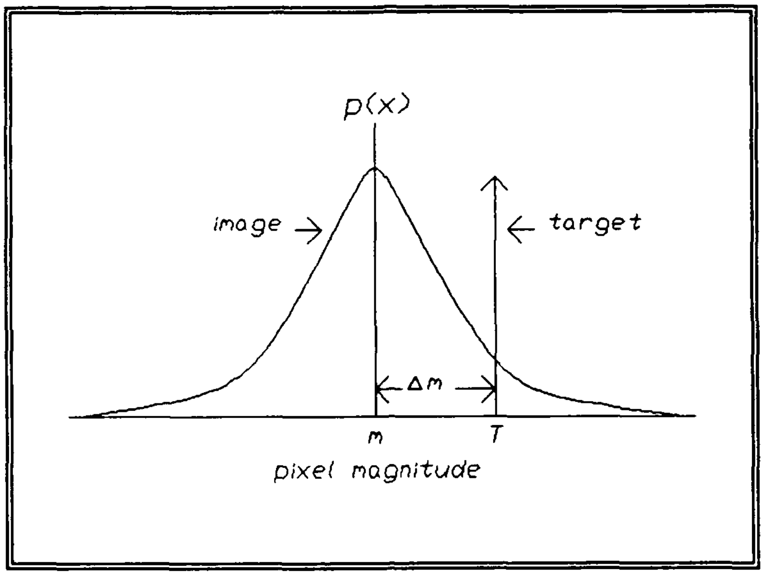

Figure 5-1. Background PDF Versus Imbedded Target

target intensity T is notable when compared to the background process. Therefore prior to processing with the adaptive PEF the background process mean is removed. This results in a new target T^* with an intensity of Δm.

During the modelling phase the coefficients of the reduced order model are adapted to fit the image at each pixel location. The error in the fit is relatively small in the locations occupied by actual background pixels. This also holds true for any pixel value that closely resembles a background pixel; a target with a small Δm is more difficult to distinguish from the background than one with a larger Δm. The value of the error process at the target location is

$$e^*(n,m) = T^*(n,m) - \sum_{\substack{\ell=0 \\ (\ell,k)\neq(0,0)}}^{P-1} \sum_{k=0}^{Q-1} h(\ell,k)x(n-\ell,m-k)$$

(5-1)

The term $e^*(n,m)$ represents the error due to the presence of the target. This term may be rewritten as a combination of two error terms. The first term represents the error that would be result from predicting the background in the absence of a target. That is,

$$e(n,m) = x(n,m) - \sum_{\ell=0}^{P-1} \sum_{k=0}^{Q-1} h(\ell,k)x(n-\ell,m-k)$$

(5-2)

The other term is the difference between the target pixel and the missing background pixel.

$$e_T(n,m) = e^*(n,m) - e(n,m) = T^*(n,m) - x(n,m)$$

(5-3)

The resultant error process $e^*(n,m)$, separated in this manner, can be compared to the background error process. The process $e(n,m)$, assuming complete decorrelation of the background, can be completely characterized by its variance. On the other hand the error process $e_T(n,m)$ possesses a mean which will be non-zero and is given by

$$E\{e_T(n,m)\} = \Delta m = T^*(n,m)$$

(5-4)

The variance of error process $e^*(n,m)$ is defined as

$$\mathrm{Var}\{e^*(n,m)\} = E\{e^*(n,m)^2\} - E\{e^*(n,m)\}^2 \tag{5-5}$$

After using (5-1) through (5-4) in (5-5) and simplifying the variance of the target error process becomes[1]

$$\mathrm{Var}\{e^*(n,m)\} = \sigma_x{}^2(n,m) - \sigma_e{}^2(n,m) \tag{5-6}$$

Intuitively this means that the ability to recognize the target is linked to the contrast between the background random process and the deterministic target. These two error processes present in a prediction error image may be viewed as in Figure 5-2. It can be seen that the magnitude of the mean difference between the target pixel and the anticipated background pixel determines the separation between the two density functions. It must be kept in mind that the density function is for the local area since the filter is spatially adaptive. Thus it is the difference between the local area mean and the target intensity that constitutes (Δm) the difference between the two density functions. The factor is evident in an image with multiple background processes. The solution presented was for a homogenous (single) background process. In an image with multiple background processes the

[1] Note that as previously discussed x(n,m) and therefore e(n,m) has a mean of zero.

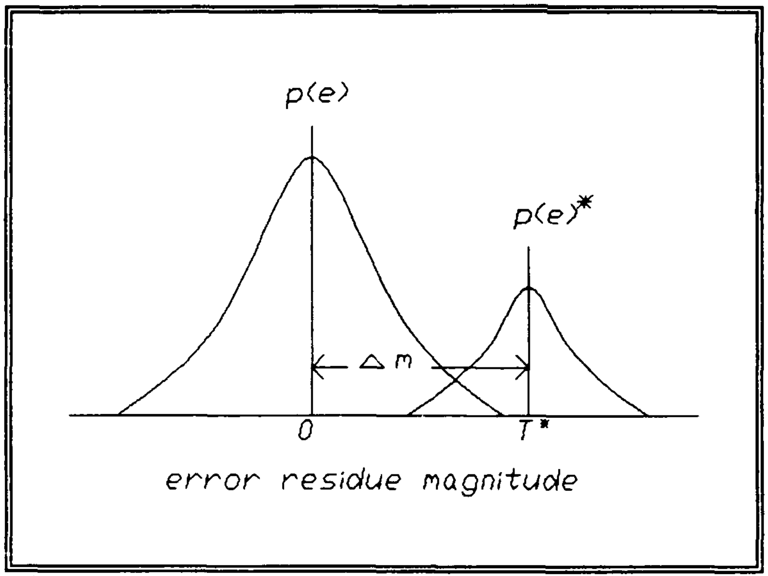

Figure 5-2. Background Error Process Versus Target Error Process

removal of the image mean does not necessarily produce a local process PDF with a mean of zero. A separation between the local background process and the target process still results but it is now with reference to the local process mean.

The form of the inverse filter as mentioned previously is a prediction error filter (PEF). This filter has a finite impulse response. It follows that the effect of a target on the error process is not limited to the pixel where the target is located. The PEF used in this thesis is rectangular with a finite impulse response of 3x3 pixels in dimension. If the background process is stationary within this region, the error process can be defined as follows. The error at each location within the impulse response support region is

$$e^*(n+i,m+j) = \sum_{\substack{\ell=0 \\ (\ell,k)\neq(i,j)}}^{2} \sum_{k=0}^{2} a(\ell,k;n+i,m+j)\; x(n+i-\ell,m+j-k) + a(i,m;n+i,m+j)T^*(n,m)$$

(5-7)

where a(ℓ,k;n+i,m+j) are the filter coefficients

$$a(\ell,k;n+i,m+j) = \begin{cases} 1 & (\ell,k) = (0,0) \\ -h(\ell,k) & 0\leq\ell\leq2,\ 0\leq k\leq2 \end{cases}$$

(5-8)

The additional indices (n+i,m+j) are to indicate that these coefficients are spatially variant. Using (5-3), (5-7), and (5-8) and concatenating the respective components into column vectors, the error process in the support region may be written as

$$\underline{e}^* = \underline{e} + e_T(n,m)\underline{a}$$

(5-9)

Now assume that the filter is at steady state within the region. Since the background has mean zero, $\underline{e}$ has mean zero. Then the mean of the error process at each location within the region in vector form follows from (5-4) and (5-9)

$$E\{\underline{e}^*\} = \Delta m\underline{a}$$

(5-10)

Since the error process is uncorrelated, its covariance matrix $\mathbf{K}_e$ is diagonal and follows from (5-6) and (5-8). The diagonal elements, k(i,j), of the covariance matrix are defined as

$$k_e(i,j) = \begin{cases} \sigma_x^{\ 2} - \sigma_e^{\ 2} & i=j=0 \\ a(i,j)^2\sigma_x^{\ 2} + \sigma_e^{\ 2} & i=j\neq 0 \end{cases} \tag{5-11}$$

The purpose for the preceding derivation of the error process in the impulse region is twofold. First, it provides some additional information which could be used for methoas of detection more oriented to hypothesis testing or significance testing of multiple pixel samples. Second, it shows that some significant error residual will occur in the immediate area of the target in the impulse response support region. The error is the result of the adaptive process that attempts to minimize the error variance. It seems reasonable that the greatest power in the error residual in the impulse response support region will be concentrated in the pixels which are the nearest neighbors of the target. This concept can be used in grouping suspected targets (results of the significance test) and for track file reduction.

B. SIGNIFICANCE LEVEL

The concept of significance testing was discussed in Chapter II. Recall that a significance test is based on a null hypothesis H_0 whose distribution is known. It is against this hypothesis that all other events are tested. The null hypothesis H_0 to be used in this section is that of the error residual process. If the background process is gaussian then the error process is also gaussian. In the preceding section the error process was discussed. The error process due to the background was defined in (5-2) and the assumption was made that the process e(n,m) was completely uncorrelated. The probability density function for this process can thus be written as

$$p(e(n,m)) = \frac{1}{(2\pi)^{1/2}\sigma_e(n,m)} \exp\left[\frac{-1}{2}\frac{e(n,m)^2}{\sigma_e(n,m)^2}\right] \tag{5-12}$$

The term $\sigma_e(n,m)$ is indexed to reflect its dependence on the local statistics of the image. This density function can be used to define the significance level.

Since the prediction error residual represents the unpredictable portion of the background it contains all of the pertinent information, without redundancy, necessary for making decisions. For this reason significance testing will be performed on the prediction error residual. The ordered set of all such residuals will be referred to as the prediction error image.

It is desirable to set the significance level in such a way that little or no adjustment is needed over the entire image. To start with, a decision is made to reject H_0 (i.e., to consider that a target may be present) whenever the magnitude of the error falls within a critical region. The lower boundary of this region is defined by the maximum acceptable two tailed probability of a type I error, P (reject H_0 given H_0 is true) (see Figure 5-3). The computed probability of a type I error, based on the null hypothesis, will be called α. Associated with this probability is a level B which defines the critical region (see Figure 5-3).

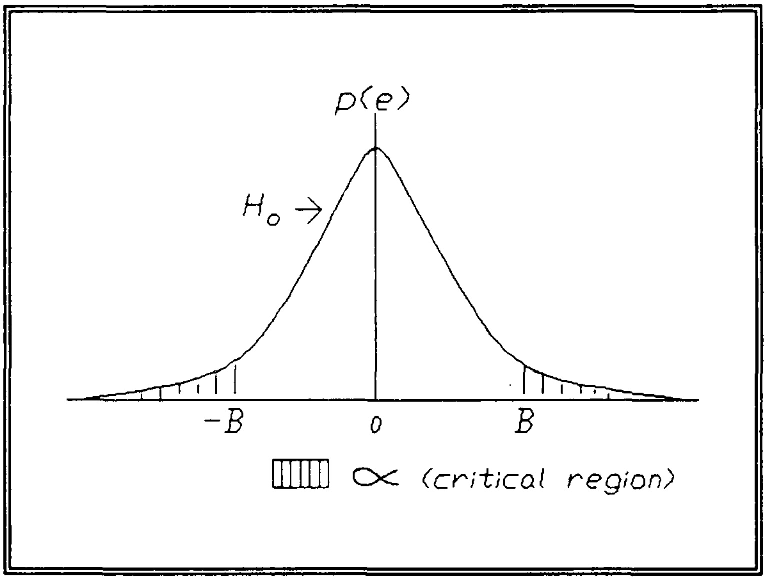

Figure 5-3. Diagram of Significance Testing Terminology

Using these last two ideas the significance level α may be written as

$$\alpha = 2\int_{B}^{\infty} \frac{1}{(2\pi)^{1/2}\,\sigma_e(n,m)} \exp\left[-\frac{1}{2}\,\frac{e(n,m)^2}{\sigma_e(n,m)^2}\right] de(n,m) \tag{5-13}$$

and the corresponding critical region is defined by

$$|e| > B \tag{5-14}$$

It can be seen that (5-13) and (5-14) depend on the local statistics. If a fixed threshold B (see Figure 5-3) is chosen, the probability α would fluctuate with the changes in variance $\sigma_e^2(n,m)$. This is undesirable. The probability α is equal to total area under the PDF within the critical region. The goal is to find a threshold which will produce a constant α independent of the local statistics.

A solution is to map the error distribution into a normalized distribution with the mapping

$$z = \frac{e(n,m)}{\sigma_e(n,m)} \tag{5-15}$$

This mapping results in a unit variance normal distribution in the variable z and the critical region is mapped into the region

$$|z| > A \qquad \text{where } A = \frac{B}{\sigma_e{}^2(n,m)} \tag{5-16}$$

The probability α is determined by

$$\alpha = 2\int_{A}^{\infty} \frac{1}{(2\pi)^{1/2}} \exp\left[-\frac{1}{2}z^2\right]dz$$

(5-17)

A depiction of the mapped error residual PDF is shown in Figure 5-4.

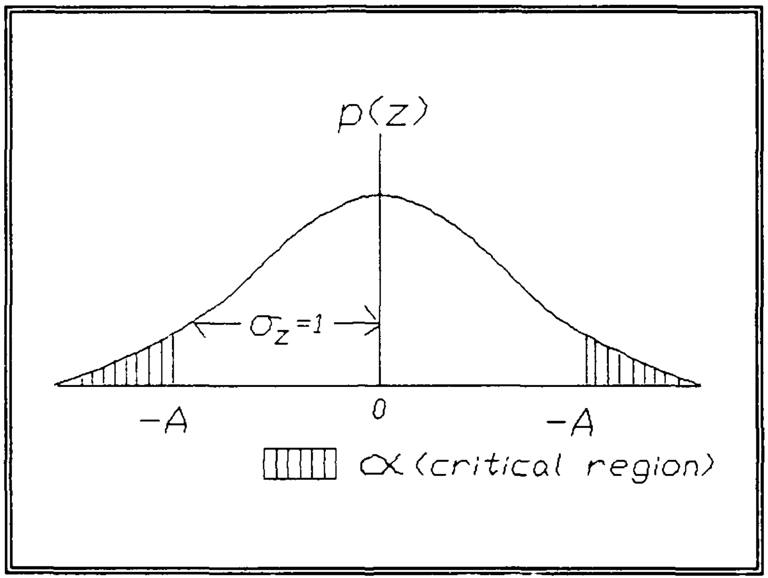

Figure 5-4. Thresholding in the Normalized PDF Space

Now observe that a choice of a threshold value of A=constant produces a constant probability α. The test (5-16) can thus be expressed, using (5-15) as

$$\frac{e(n,m)^2}{\sigma_e(n,m)^2} > A^2 = \text{constant}$$

(5-18)

From this it can be seen that a constant alarm rate (α) can be achieved with a constant threshold. The implication of this is that the statistic on the left hand side of (5-18) is independent of the variations in the background or error residual image. Ideally then, a single threshold would be suitable for the entire image. This constant threshold may be found experimentally by adjusting the threshold in the prediction error images to produce on the average a specified percentage of false alarms (pixels which exceed the threshold).

The preceding discussion defined the statistic to be tested and showed that the threshold may be set to any constant which produces the desired constant false alarm rate. The terms e(n,m) are computed using either the LMS or RLS method. In addition the term $\sigma_e^2(n,m)$ needs to be computed. In the RLS algorithm this estimate was formed by (4-39). During the LMS process the estimate of $\sigma_e^2(n,m)$ is computed from the difference in the unbiased spatial averages of the input $x(n,m)^2$ and the prediction $\hat{x}(n,m)^2$. To compute this an unbiased estimate of the correlation matrix ($\mathbf{R}_x$) was used in conjunction with a column vector of the PEF coefficients as discussed earlier. The following equation defines the estimate of the error variance and is computed at each pixel location.

$$\sigma_e^2(n,m) = \underline{a}^T R_x \underline{a}$$

(5-19)

This completes the description of the significance test. A block diagram of the process through the testing stage is shown in Figure 5-5.

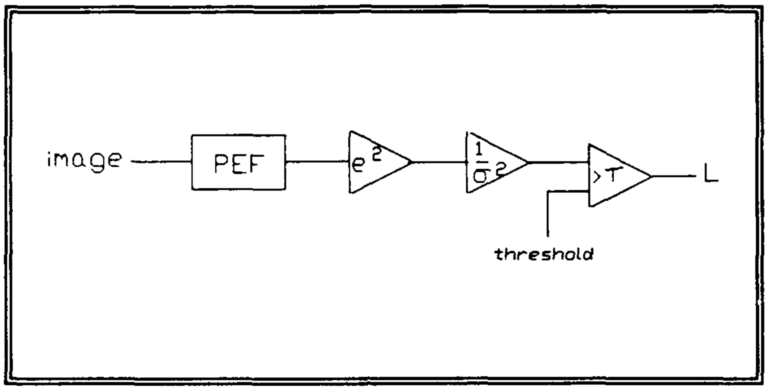

Figure 5-5. Diagram of Detection Process

The testing can be carried out at the same time the image is being whitened. This reduces the requirements for additional storage. It was mentioned earlier that the significance testing should be followed by a method for managing the resultant target list. This will not be addressed in this thesis. However a possible tracking algorithm is covered in the next chapter.

C. SIMULATION RESULTS

The simulations conducted using the processing algorithms developed up to this point were on images with homogenous backgrounds. The image mean was computed and removed prior to processing; however, the local mean was not removed. The background of the image was generated from one of several models of the form

$$x(n,m) = \sum_{\substack{\ell=0 \\ (\ell,k)\neq(0,0)}}^{2} \sum_{k=0}^{2} h(\ell,k)x(n-\ell,m-k) + w(n,m) \tag{5-20}$$

where $h(\ell,k)$ represents a fixed set of filter coefficients and $w(n,m)$ is a gaussian white noise source with a variety of variances. After preparing the image background, targets were placed at random locations within the image. The significance testing results with the various models and driving source noise were consistent throughout. The dimensionality of the PEFs used were 3x3 pixels; this was more than adequate for the 2x2 model. It should be realized that in other cases the size of the PEF required to whiten a specific image may be ıarger or smaller than 3x3.

The specific results presented in this chapter are for the background model in (5-20) with the coefficients $h(1,0)=0.5$, $h(0,1)=0.8$, $h(1,1)=-0.4$ and $\sigma_w^2 = 2.0$. The imbedded targets were set to specific dB levels relative to the local variance in the 9x9 pixel region in which they were to be imbedded. This has been designated TBR (target to background ratio). Once the TBR was set, the mean of the 9x9 region was added to the target. The target then replaced the pixel at the center of the 9x9 region. The purpose of this was to test the capability of the whitening and detection algorithms on targets that differed by very small amounts from their neighbors. Some simulations were conducted for images with imbedded targets corrupted by additive white noise. The level of noise is specified as the BNR (background to noise ratio). The averaged results for multiple simulations at each parameter setting are shown in the Appendix.

Plots of the results of processing identical images with the LMS algorithm and the RLS algorithm are shown in Figures 5-6 and 5-7 respectively. These plots are for the conditions of no noise and a TBR of 5 dB with the threshold set to 11.4 and ten targets imbedded in the image. Figures 5-8 and 5-9 are the same images with imbedded targets of the same TBR as in Figures 5-6 and 5-7. However the images were corrupted with additive white noise. The BNR for both is 10 dB. These images were, respectively, processed with the LMS and RLS algorithms and the threshold was set to 11.4.

Earlier in the chapter a model of the two error processes operating in a prediction error image was presented. It can be seen in Figure 5-2 that if the separation between the two processes is large, the threshold can be set to achieve a high probability of target detection while maintaining a low probability of false alarms. Refer to Tables 6 and 14 in the

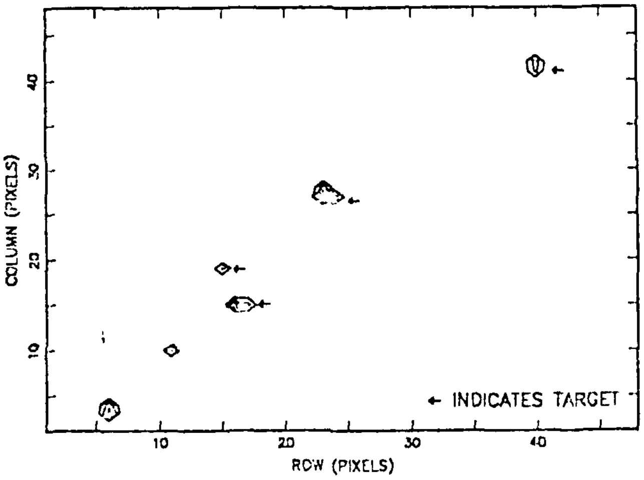

Figure 5-6. Significance Test Results on an LMS Processed Image (No Noise)

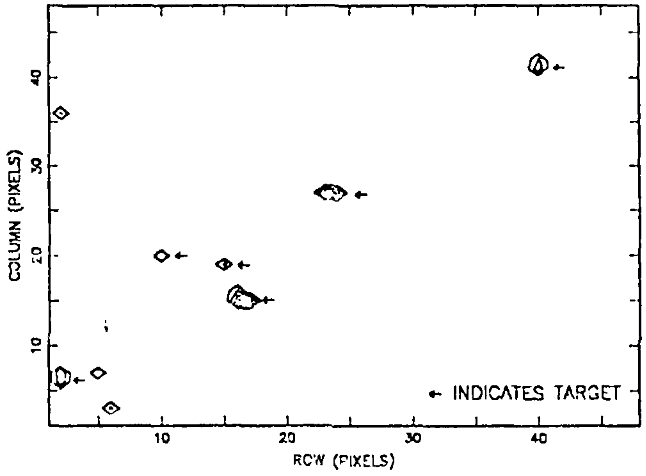

Figure 5-7. Significance Test Results on an RLS Processed Image (No Noise)

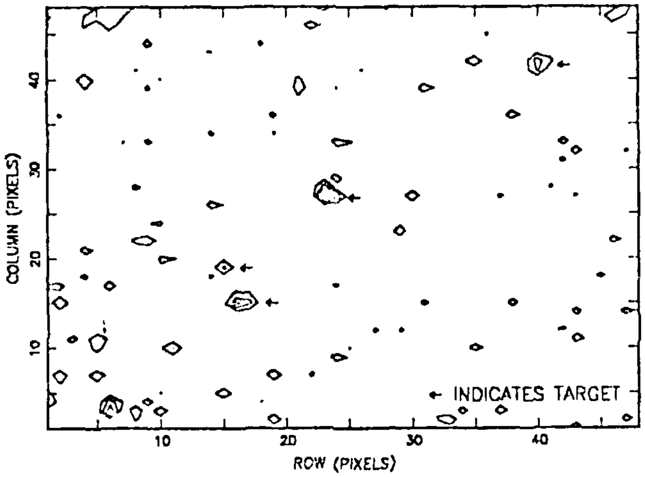

Figure 5-8. Test Results on a LMS Processed Image (BNR=10dB)

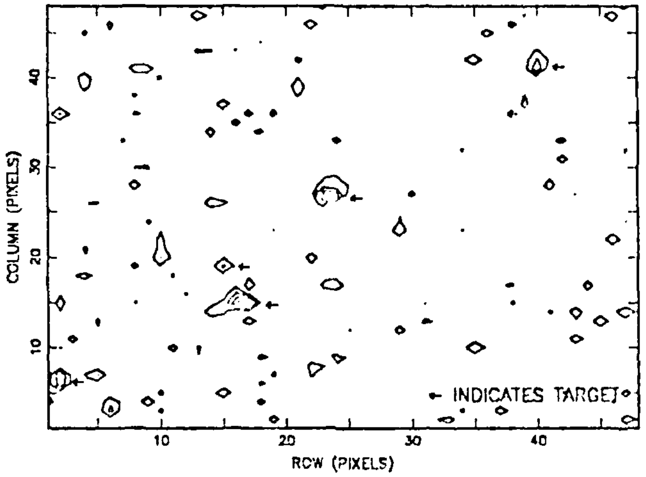

Figure 5-9. Test Results on a RLS Processed Image (BNR≈10dB)

Appendix and recall the discussion (Chapter III) on the excess MSE resulting from use of the LMS algorithm. The RLS algorithm by its nature does not suffer from this type of excess MSE. Thus it would be expected that the whitened background process PDF would be broader for the LMS than for the RLS. This result can be seen in Table 6 and 14 by comparing the false alarm probabilities at which 100%

detection occurs. The false alarm probabilities essentially are a measure of the tail area in both tails of the background error process. It is seen from the tables that the broadened LMS error process adds an additional 9.25% to the tail area (false alarm probability). In another set of results, Tables 3 and 11, the increased false alarm probability is seen to be approximately 6.5%.

The effect of the increased background error variance can also be seen in Figures 5-6 and 5-7. In general from the tables it may be seen that at the same significance level the number of false alarms present remains approximately constant for both LMS and RLS. This result supports the proposed thresholding method and independence of the test statistic in (5-18). However the number of true targets found to be above the threshold is smaller for the LMS method than for the RLS method.

The effect of the target error process in the impulse response support region can be seen in Figures 5-6 and 5-7. The scan direction for these figures was a row-wise raster pattern. The broadening of the error residual in the vicinity of the detected targets can be seen in the figures. It can be seen that the largest error residual occurs at the locations within one pixel of the true target location.

It is possible to use the spread in the thresholds as an indicator of the spread in the target process. Bear in mind that comparison between tables is subject to the effects of squaring the error in the test statistic. Also note in the tables the translation of the distribution of the thresholds is due to increases in the target magnitude. These points lend support to the assumed model.

The preceding comparisons were the results of testing without added noise. It would be expected that adding noise would tend to increase the variance of the background error process while narrowing the target error process. The latter is due to the reduced ability of the adaptive filter to follow the changing background process. However in Tables 3 through 5, for the LMS process with light noise (BNR=30 dB) the performance is actually better than with no noise. In adaptive gradient search algorithms, overlapping input data sets introduces correlation between the filter coefficient error vector and the data under the prediction mask. This creates an error bias in such algorithms as LMS [Ref. 13]. The addition of a small amount of white noise tends to reduce the correlation therefore decreasing MSE. This would decrease the width of the background error process and increase the variance of the target process. In Table 5 the beneficial effect is overcome by the increased variance of the target and background processes. Note that in Tables 11 through 13 for the RLS method, the background error process broadens as expected and as indicated by the increased false alarm probability. In Table 12 some benefit is seen in light noise (due to target process shifting), but considerably less than occurred in the LMS case. A visual example of the effect of noise on the two methods is shown in Figures 5-8 and 5-9.

From the plots and tables it can be seen that the RLS algorithm is superior to the LMS algorithm particularly in the area of dim targets and in the presence of noise. Of course the RLS algorithm is computationally more intensive than the LMS algorithm. The LMS computational advantage however may be offset by the additional computational burden of processing the increased number of false alarms.

A point of particular interest is a region which may be called the "knee" of the significance testing process. This is the point at which the false alarm rate approximately doubles for a 10% increase in detection probability. This point appears in all of the tables and occurs at approximately 80% detection probability for an LMS whitened image and 90% detection for RLS whitening. While these are not limits on the detection process they may be useful as figures of merit when comparing methods of whitening the image.

D. CONCLUSIONS

It has been proposed that the decorrelation of the background process results in a separation of the target process from the background process. The experimental data supports this concept. Detection of targets at low false alarm rates requires an adaptive PEF that produces the smallest possible error variance. Thus it is not surprising in the comparison of the LMS and RLS methods that the RLS results are better. The false alarm rates using this detection method are relatively low for the detection rates achieved (see Appendix). This is due to the ability of the filter to adapt to local changes in the image. This in turn results in prediction coefficients which more closely model the local process and leads to a more complete decorrelation of the image by the PEF.

The addition of white noise to images degrades the algorithm performance but not to the extent that might be anticipated. The noise broadens the background error process density function thus adding more false alarms for a given threshold setting. However the target error process variance is reduced and offsets some of the effect. Light noise (BNR=30 dB) was found to increase the detection rate, particularly for LMS. Inevitably severe degradation occurs as the noise level increases above 10 dB.

Significance testing showed some capability for detecting targets with TNR's near 0 dB. However the false alarm rate at that noise level may be higher than can be tolerated. The conclusion is that the methods used show considerable promise and capabilities in highlighting dim targets with TBR's of (1 to 2 dB) and in the presence of light to moderate noise levels.

VI. TARGET TRACKER

A. GENERAL

Throughout this thesis it has been assumed that the targets to be located are pixel or sub pixel in size. It has been shown that with the methods addressed in this thesis that target pixels can be identified. However this method also produces false alarms. The number of false alarms has been shown to be contingent on the significance level selected. If the significance level is set to achieve higher probabilities of target detection, the number of false alarms increase. Follow-on processing is required to reduce the number of false alarms and to track the targets spatially in successive images.

The tracking method proposed in this chapter is based on the multi-dimensional pulse tracker proposed by Therrien in [Ref. 26]. The tracker to be developed links the two objectives stated in the previous paragraph through the target motion. Three possibilities for the motion of the suspected target are no motion, uncorrelated, and correlated. Targets without motion will persist at the same location in the image through time. Uncorrelated motion is spatial displacement between images which is entirely random and unpredictable.

Correlated motion is random spatial displacement; however the target motion exhibits a trend in direction and rate of movement.

A tracker can be used as an observer, providing movement information that can be used to categorize the targets. This information and *a priori* knowledge of the target dynamics can be used to reduce the target list. This is accomplished by comparing the observation to a specific profile of target dynamics. If the observation fits the profile, the target is retained in the target list; otherwise it is dropped. This thesis will not discuss the logic necessary to reduce the target list. The preceding was presented to indicate a possible means of using the tracker as a additional source of target information. Later in the chapter it will be shown that some target dynamic information can be incorporated into the tracker to help in tracking.

Processing described in the previous chapters provides the location of possible targets and access to two types of image. These images are the actual observed image and the prediction error image. In these images the target has a deterministic shape which can be exploited in the construction of a tracker. It is assumed, as before, that the target in the observed image is approximately one pixel in size and has a non-zero magnitude. It was subsequently shown that the target, after whitening with the PEF, experiences point spreading due to the filter's impulse response. An image containing a target with intensity profile s(n,m) can be modeled as

$$r(n,m;t) = s(n+\zeta_n,m+\zeta_m;t) + w(n,m) \tag{6-1}$$

where r(n,m;t) is the observed intensity at a pixel (n,m) in an image at time t. The term $s(n+\zeta_n,m+\zeta_m;t)$ represents the target which should have occurred at (n,m) but instead occurred at $(n+\zeta_n,m+\zeta_m)$. The terms ζ_n and ζ_m are the position error in the target location at time t. This can be viewed as a translation of the target from the expected location due to motion. The term w(n,m) is a additive noise term. The observed image in the absence of a target can be modeled as

$$r(n,m;t) = w_c(n,m) + w(n,m) \tag{6-2}$$

where $w_c(n,m)$ is colored noise (background or incompletely whitened background) and w(n,m) is additive white noise. The tracking problem is simplified if the term $w_c(n,m)$ is near zero or if the target power is considerably larger than the colored noise component power. In either case the colored noise component can be ignored. For these reasons it was decided to perform the tracking in the prediction error image.

During the background whitening stage the PEF coefficients are continually adjusted to produce the minimum error variance. This produces a near minimal phase filter and results in most of the impulse response power occurring near the target location. With these considerations the target intensity s(n,m) in (6-1) was modeled as a spread gaussian pulse of the form[2]

$$s(n,m) = \begin{cases} C \exp\left[-\frac{1}{2} \frac{n^2+m^2}{\sigma_s^2} \right] & 0 \leq |n| \leq \frac{N-1}{2}, \\ & 0 \leq |m| \leq \frac{M-1}{2} \\ 0 & \text{elsewhere} \end{cases} \tag{6-3}$$

The term σ_s^2 controls the spread of the modeled signal. For a given PEF impulse response support area, σ_s is fixed. Experimentally the best value was found to correspond to be a radius around the target in which approximately 90% of the impulse response power occurs. For the adaptive 3x3 PEF this typically occurred at a radius of about one pixel. The scale factor C is used to match the signal magnitude and polarity. Polarity is positive if the target intensity is above the local image mean and negative if it is below. In order to match the model to the target signal, an estimate of the magnitude and polarity of the suspected targets are computed by the tracker when they are passed by the detection algorithm.

[2] Although the true target intensity profile may not be Gaussian, this form is convenient. It will be seen later that this tracker is quite robust with respect to the detailed assumptions about target shape.

The detection algorithm identifies a possible target to the tracker. The tracker then extracts, from the prediction error image, a grouping of pixels centered on the target pixel. This will be referred to as the target tracking window and its center point will be called the centroid. A square tracking window is used with the method developed in this thesis. The window should be large enough to include most of the filter impulse response. Since two targets of equal magnitude falling completely in the tracker window would be treated as if there were only one, the window maximum dimension is based on the minimum distances between adjacent tracks.

Once the target has been identified and the target location has been passed to the tracker, track processing becomes autonomous. The tracker will compute successive estimates of the target location in future images. With each estimate, the new tracking window centroid coordinates are computed and the tracking window is repositioned at the new centroid. This autonomous tracking method permits parallel processing for detection and tracking but may require a separate prediction error image storage buffer for the tracker. The buffer is required if the processing speed of the whitening and detection processor is faster than that of the tracking algorithm.

Additional logic would be needed to compare subsequent targets, identified by the detection processor, with existing tracks. Although not specifically addressed in this thesis some of the possible features of the logic would be: same target discrimination, split and merge of tracks, lost track identification, and so on. Ideally the complete tracking and logic package would produce a steady state number of false alarms tracks. This occurs when the number of new tracks started equals the number of old tracks terminated. A block diagram of the relation between the processing blocks in the detection and tracking system is shown in Figure (6-1).

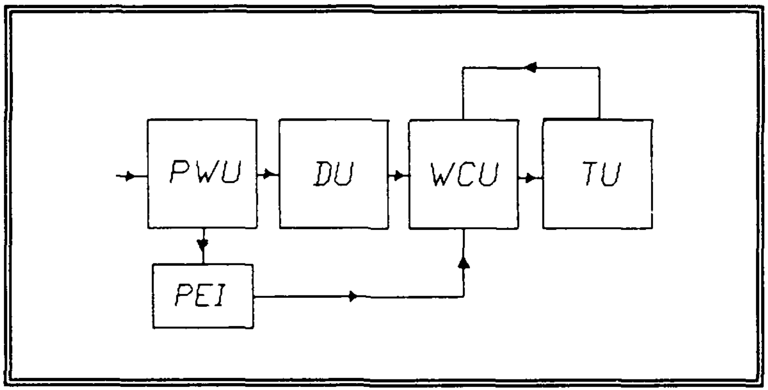

Figure 6-1. Detection and Tracking System Block Diagram

The overall system, less connective logic, is a Pre-Whitening Unit (PWU), a Detection Processing Unit (DPU), a Tracking Window Control Unit (WCU), and the Tracking Unit (TU). The tracking window control unit and the tracking unit comprise what has been referred to as the "tracker." The WCU controls the positioning of the tracking window based on the estimates in the target position provided by the tracking unit. The tracking window is then repositioned so that the centroid is at the estimated target location in successive images.

B. TRACKER

In the derivation of the tracker to follow the target position error (ζ_n,ζ_m) in (6-1) is assumed to be a zero mean gaussian random process. Further it is assumed that the components of the error in the n and m direction are independent and that the noise in the models (6-1) and (6-2) is a zero-mean white gaussian random process.

The origin of the coordinate system (n,m) is the centroid of the tracking window. The modeled target image (6-1) is located at an assumed position (ζ_n,ζ_m) which is to be estimated. A diagram of the tracking window and its terminology is shown in Figure 6-2.

Initially an estimator for the position error of a single target pixel will be developed. This will then be extended to all the pixels in the tracking window. To begin with, some notation will be defined. Let the vector ζ be defined as

$$\zeta = \begin{matrix} \zeta_n \\ \zeta_m \end{matrix} \tag{6-4}$$

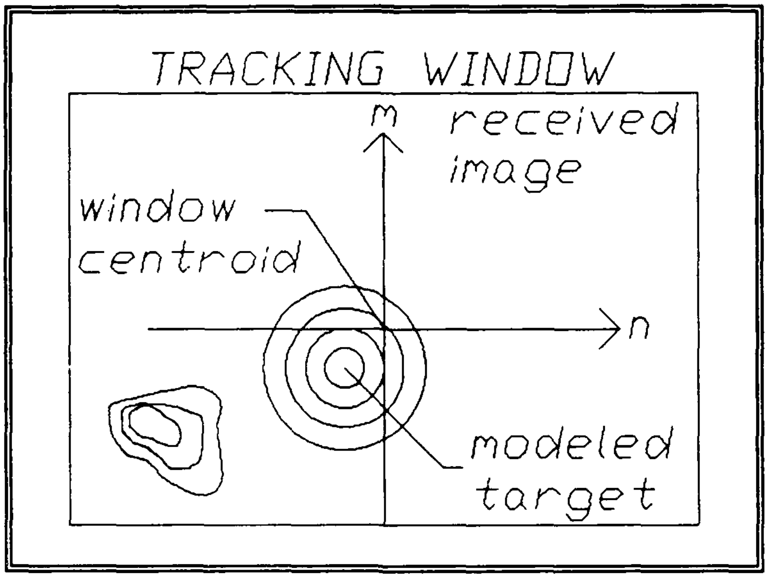

Figure 6-2. Diagram of Tracking Terminology

Further let $\underline{z}$ be the vector process of position errors in the target intensity at a pixel location in the tracking window, that is

$$\underline{z}^T = [\zeta_n^1 \zeta_n^2 \cdots \zeta_n^k \mid \zeta_m^1 \zeta_m^2 \cdots \zeta_m^k] \tag{6-5}$$

Thus the vector $\underline{z}$ is comprised of the position error for a given pixel in the tracking window in each of k successive images.

The vector $\underline{r}$ is similarly defined as the sequence of image intensities at a given pixel location in the tracking window, that is

$$\underline{r} = \begin{bmatrix} r(n,m;t^1) \\ r(n,m;t^2) \\ \cdot \\ \cdot \\ \cdot \\ r(n,m;t^k) \end{bmatrix}$$

(6-6)

Finally the target vector $\underline{s}(\underline{\zeta})$ is a column vector containing the sequence of target intensity values present at the same location (n,m) in the tracking window, that is

$$\underline{s}(\underline{\zeta}) = \begin{bmatrix} s(n+\zeta_n^1,m+\zeta_m^1;t^1) \\ s(n+\zeta_n^2,m+\zeta_m^2;t^2) \\ \cdot \\ \cdot \\ \cdot \\ s(n+\zeta_n^k,m+\zeta_m^k;t^k) \end{bmatrix}$$

(6-7)

The index t^i, for i=1,2,3...k, is the time index for the k image frames considered in estimating the position error. Each pixel intensity is a sample of the continuous signal (6-3) within the tracking window. The observed target is assumed to be deterministic but subject to a random position error and additive white noise. The situation is depicted in Figure 6-2.

The goal of the tracking algorithm is to predict the position error in the target in succeeding images. One method of accomplishing this is to use MAP estimation. Given the statistical models for the position error process and assuming for the moment that the form of the target image process is known and given by (6-3) then it is possible to compute the a posteriori conditional density function $p(\underline{z}|\underline{r})$. From this density it is possible to determine the best estimate $\hat{\underline{z}}$ in the MAP sense. It will be shown that future estimates of $\underline{\zeta}$ can be formed recursively from this estimate. [Ref. 26] Using Bayes rule the a posteriori probability density $p(\underline{z}|\underline{r})$ can be evaluated from

$$p(\underline{z}|\underline{r}) = \frac{p(\underline{r}|\underline{z})p(\underline{z})}{p(\underline{r})} \tag{6-8}$$

Thus using (6-1) the conditional probability density function $p(\underline{r}|\underline{z})$ is

$$p(\underline{r}|\underline{z}) = \frac{1}{(2\pi)^{k/2}|\mathbf{R}_w|^{1/2}}\exp[- \frac{1}{2} (\underline{r}-\underline{s}(\underline{\zeta}))^T\mathbf{R}_w^{-1}(\underline{r}-\underline{s}(\underline{\zeta}))] \tag{6-9}$$

where $\mathbf{R}_w$ is the covariance matrix of the additive white noise process. The a priori probability density $p(\underline{z})$ based on the assumptions stated earlier is

$$p(\underline{z}) = \frac{1}{(2\pi)^{k/2}|\mathbf{R}_z|^{1/2}}\exp[- \frac{1}{2} \underline{z}^T\mathbf{R}_z^{-1}\underline{z}] \tag{6-10}$$

where $\mathbf{R}_z$ is the covariance matrix of the position error in the n and m directions. Because of the assumed independence of ζ_n and ζ_m this matrix can be partitioned as

$$\mathbf{R}_z = \left[\begin{array}{c|c} \mathbf{R}_z^n & 0 \\ \hline 0 & \mathbf{R}_z^m \end{array}\right]$$

(6-11)

In order to find an MAP estimate we need to find the value of $\underline{z}$ that maximizes (6-8). Since the denominator is not a function of $\underline{z}$ it is sufficient to find $\underline{z}$ to maximize the numerator. If we substitute (6-9) and (6-10) into the numerator of (6-8), take the logarithm, and drop the constant term we are left with

$$C(\zeta) = -\frac{1}{2}[(\underline{r}-\underline{s}(\zeta))^T \mathbf{R}_w^{-1}(\underline{r}-\underline{s}(\zeta)) + \underline{z}^T \mathbf{R}_z^{-1}\underline{z}]$$

(6-12)

To determine $\hat{\underline{z}}_{opt}$ (the optimum estimate of $\underline{z}$) the derivative of (6-12) is taken with respect to the position error vector $\underline{z}$ and the result is set equal to the zero vector.

$$\frac{\partial}{\partial \underline{z}^T} C(\zeta) = \left[\frac{\partial \underline{s}^T(\zeta)}{\partial \underline{z}}\right] \mathbf{R}_w^{-1}(\underline{r}-\underline{s}(\zeta)) - \mathbf{R}_z^{-1}\underline{z} = 0$$

(6-13)

Thus $\hat{\underline{z}}_{opt}$ is defined implicitly by

$$\hat{\underline{z}}_{opt} = R_z \left[\frac{\partial \underline{s}^T(\zeta)}{\partial \underline{z}}\right] R_w^{-1}(\underline{r}-\underline{s}(\zeta))$$

(6-14)

where the partial derivative term is a matrix whose components are $[\partial \underline{s}_i / \partial \underline{\zeta}_i]$. The signal present in any given image is subject to the position error in only that image. Therefore the partial derivative matrix has the form

$$\partial \underline{s}(\underline{\zeta}) = \left[\begin{array}{cccccc} \frac{\partial s(\zeta^1)}{\partial \zeta n^1} & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & \frac{\partial s(\zeta^2)}{\zeta n^2} & & 0 & \cdot\ \cdot\ \cdot & 0 \\ \cdot & & \cdot & & & \\ \cdot & & & \cdot & & \\ \cdot & & & & \cdot & \\ 0 & \cdot & \cdot & \cdot & & \frac{\partial s(\zeta^k)}{\partial \zeta n^k} \end{array}\right|$$

$$\left.\begin{array}{cccccc} \frac{\partial s(\zeta^1)}{\partial \zeta m^1} & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & \frac{\partial s(\zeta^2)}{\partial \zeta m^2} & & 0 & \cdot\ \cdot\ \cdot & 0 \\ \cdot & & \cdot & & & \\ \cdot & & & \cdot & & \\ \cdot & & & & \cdot & \\ 0 & \cdot & \cdot & \cdot & & \frac{\partial s(\zeta^k)}{\partial \zeta m^k} \end{array}\right]$$

(6-15)

At this point it is beneficial to return to Equation (6-14). Now recall that the noise, w, was assumed to be white. Therefore the covariance matrix $\mathbf{R}_w$ is a diagonal matrix with components equal to the noise power N_o. Thus if we take advantage of this and the partitioning of (6-15), the estimate for $\underline{z}$ can be written as

$$\hat{\underline{z}}_n(n,m) = \mathbf{R}_z^n \left[\frac{\partial s^T(\underline{\zeta})}{\partial \underline{\zeta}_n} \right] \frac{1}{N_0} \quad (\underline{r} - s(\underline{\zeta}))$$

(6-16)

and

$$\hat{\underline{z}}_m(n,m) = R_z^m \left[\frac{\partial s^T(\underline{\zeta})}{\partial \zeta_m} \right] \frac{1}{N_0} \quad (\underline{r} - s(\underline{\zeta}))$$

(6-17)

where $\hat{\underline{z}}_n(n,m)$ and $\hat{\underline{z}}_m(n,m)$ are the position error estimates in the n and m directions, respectively, and are optimal in the MAP sense.

Now define vectors $\underline{\alpha}(n,m)$ and $\underline{\beta}(n,m)$ as

$$\underline{\alpha}(n,m) = \left[\frac{\partial s^T(\underline{\zeta})}{\partial \underline{\zeta}_n} \right] \frac{1}{N_0} \quad (\underline{r} - s(\underline{\zeta}))$$

(6-18)

and

$$\underline{\beta}(n,m) = \left[\frac{\partial s^T(\underline{\zeta})}{\partial \underline{\zeta}_m} \right] \frac{1}{N_0} \quad (\underline{r} - s(\underline{\zeta}))$$

(6-19)

Then the two estimators can be written in the compact form

$$\underline{z}_n(n,m) = \mathbf{R}_z^n \, \underline{\alpha}(n,m)$$

(6-20)

and

$$\underline{z}_m(n,m) = \mathbf{R}_z^m \, \underline{\beta}(n,m)$$

(6-21)

It will be assumed for the moment that the covariance matrices $\mathbf{R}_z^n$ and $\mathbf{R}_z^m$ are known. Equations (6-20) and (6-21) then each represent a set of k scalar equations. The last equations of these sets have the forms

$$\zeta_n(n,m;p) = \sum_{i=p-k+1}^{p} R_z^n(p-i)\alpha(n,m;i)$$

(6-22)

and

$$\zeta_n(n,m;p) = \sum_{i=p-k+1}^{p} R_z^m(p-j)\beta(n,m;i)$$

(6-23)

where $\alpha(n,m;i)$ and $\beta(n,m;i)$ are the i^{th} components of $\underline{\alpha}$ and $\underline{\beta}$ defined as

$$\alpha(n,m;i) = \frac{1}{N_0}\left[\frac{\partial \underline{s}^T(\underline{\zeta})}{\partial \zeta_n{}^{i}}\right] (\underline{r}-\underline{s}(\underline{\zeta}))$$

(6-24)

and

$$\beta(n,m;i) = \frac{1}{N_0}\left[\frac{\partial \underline{s}^T(\underline{\zeta})}{\partial \zeta_m{}^{i}}\right] (\underline{r}-\underline{s}(\underline{\zeta}))$$

(6-25)

These terms are the inner product of the gradient of the signal multiplied by the difference of the observed and estimated images. These equations provide the means by which the target position estimate is updated.

Consider the form of $\underline{s}(\underline{\zeta})$ Equation (6-7) in conjunction with the estimator Equations (6-22) and (6-23). It can be seen that in order to compute the recursive estimates for the position error, k previous estimates are required. The upper limit in the summation represents the most recent image time i=p=0 representing the beginning of the track. It can be seen that the terms of (6-22) and (6-23) with indices i<p represent the track history prior to time i=p. These terms for a causal system are identically zero. At the beginning of the track the observed target location is passed to the tracker. Then the tracking window centroid is placed at the observed target location. By design the modeled target $s(n+\zeta_n, m+\zeta_m)$ is fixed at the center of the tracking window. Therefore the estimate of the position errors, ζ_n and ζ_m, at the beginning of the track is known and is incorporated in the centroid position. Thus all the essential terms are available recursively to compute the future estimates.

The terms $\zeta_n(n,m;p)$ and $\zeta_m(n,m;p)$ pertain to a particular pixel within the tracking window. To compute the new position of the tracking window centroid, these terms are averaged. Specifically, let Δ_n^{p+1} and Δ_m^{p+1} be the change in the centroid position. Then the change in the centroid position is computed from

$$\Delta_n^{p+1} = \sum_{i=p-k+1}^{p} R_z^n(p-i)\alpha_{avg}(i) \tag{6-26}$$

and

$$\Delta_m^{p+1} = \sum_{i=p-k+1}^{p} R_z^m(p-i)\beta_{avg}(i) \tag{6-27}$$

where we have defined

$$\alpha_{avg}(i) = \frac{1}{W^2} \sum_{n=-\frac{W+1}{2}}^{\frac{W-1}{2}} \sum_{m=-\frac{W+1}{2}}^{\frac{W-1}{2}} \alpha(n,m;i) \tag{6-28}$$

and

$$\beta_{avg}(i) = \frac{1}{W^2} \sum_{n=-\frac{W+1}{2}}^{\frac{W-1}{2}} \sum_{m=-\frac{W+1}{2}}^{\frac{W-1}{2}} \beta(n,m;i) \tag{6-29}$$

The results of (6-26) and (6-27) represent the predicted error in the target location in the next image. The new centroid position is computed as

$$x_{centroid}^{p+1} = x_{centroid}^{p} + \Delta_n^{p+1} \tag{6-30}$$

and

$$y_{centroid}^{p+1} = y_{centroid}^{p} + \Delta_m^{p+1} \tag{6-31}$$

The entire procedure is repeated in each succeeding image frame. In the absence of a priori knowledge of the future target position, the error in the new estimated target location is assumed to be zero. It can be seen that the only information that needs to be retained to continue the recursion is the k past values of $\alpha_{avg}(i)$ and $\beta_{avg}(j)$.

C. TRACKER DYNAMICS

The purpose of this section is to provide some insight into the dynamics of the tracker. Further a method of incorporating the target dynamics into the design of the tracker is shown.

In the following analysis the motion of the target through the sequence of images will be assumed to be uncorrelated (see section A of this chapter). This type of motion is consistent with the problem of tracking small point targets with displacement between image frames of approximately one to two pixels. Since tracking is performed separately in the horizontal (n) and vertical (m) directions, we will only consider a single direction (horizontal) in the analysis. All results shown also apply to tracking in the vertical direction.

It follows from the assumption of uncorrelated target motion, that the error in the target position in the current image frame is independent of its position error in the previous frame. As a result, the position error correlation matrix $\mathbf{R}_z$ is diagonal. The target position error estimates for the $p+1^{st}$ image (6-26) and (6-27) then become

$$\Delta_n^{p+1} = \frac{1}{N_0 W^2} \sum_{n=-\frac{W+1}{2}}^{\frac{W-1}{2}} \sum_{m=-\frac{W+1}{2}}^{\frac{W-1}{2}} \frac{\partial s(n+\zeta_n, m+\zeta_m)}{\partial \zeta_m} \underline{r}_z^n(\underline{r}-\underline{s}(\underline{\zeta}))$$

(6-32)

and

$$\Delta_m^{p+1} = \frac{1}{N_0 W^2} \sum_{n=-\frac{W+1}{2}}^{\frac{W-1}{2}} \sum_{m=-\frac{W+1}{2}}^{\frac{W-1}{2}} \frac{\partial s(n+\zeta_n, m+\zeta_m)}{\partial \zeta_m} \underline{r}_z^m(\underline{r}-\underline{s}(\underline{\zeta}))$$

(6-33)

where $r_z^n(0)$ and $r_z^m(0)$ are the values of the correlation function at lag zero. Since it was assumed that the position error process has zero mean, these terms are also the position error variances. Further note that the gradient is reduced to a scalar gradient of the modeled target image which is evaluated at each point in the tracking window. In order to simplify the notation in this section, the scalar gradient term in (6-32) will be written as $\nabla_n(n,m)$.

With Equations (6-1) and (6-32) the mean of the position error estimate in the horizontal (n) direction is found to be

$$\begin{aligned} E\{\Delta_n^{p+1}\} &= \frac{r_z^n(0)}{N_0 W^2} \sum_{\substack{n \quad m \\ (n,m)\epsilon\{W,W\}}} \sum \nabla_n(n,m) E\{r(n,m;p)-s(n,m)\} \\ &= \frac{r_z^n(0)}{N_0 W^2} \sum_{\substack{n \quad m \\ (n,m)\epsilon\{W,W\}}} \sum \nabla_n(n,m) E\{w(n,m;p)\} \\ &= 0 \end{aligned}$$

(6-34)

where w(n,m;p) is the additive zero mean white noise in the p^{th} image frame. Further the variance of the position error estimate is

$$E\{(\Delta_n{}^{p+1})^2\} = \left(\frac{r_z^n(0)}{N_0W^2}\right)^2 E\{ \sum_{n}\sum_{\substack{m\\(n,m)\epsilon\{W,W\}}} \nabla_n(n,m)(r(n,m;p)-s(n,m;p))$$

$$\sum_{\ell}\sum_{\substack{k\\(\ell,k)\epsilon\{W,W\}}} \nabla_n(\ell,k)(r(\ell,k;p)-s(\ell,k;p))\}$$

$$= \left(\frac{r_z^n(0)}{N_0}\right)^2 \frac{1}{W^2} \sum_{n}\sum_{\substack{m\\(n,m)\epsilon\{W,W\}}} \nabla_n(n,m)^2$$

$$E\{\zeta_n{}^{p+1}\} = \frac{1}{N_0}\left(\frac{r_z^n(0)}{W}\right)^2 \sum_{n}\sum_{\substack{m\\(n,m)\epsilon\{W,W\}}} \nabla_n(n,m)^2$$

(6-35)

Note that if the tracker is to follow target position changes in the image, the variance of Δ_n^{p+1} should be proportional to the variance of the change in the frame to frame position. Thus the tracker can be optimized to follow targets with a particular velocity by adjusting the term $r_z^n(0)$.

Figures 6-3 and 6-4 show plots of the target position error estimate versus the actual target position error. In Figure 6-3 the estimates are for target motion completely in the horizontal (n) direction. In Figure 6-4 the estimates are

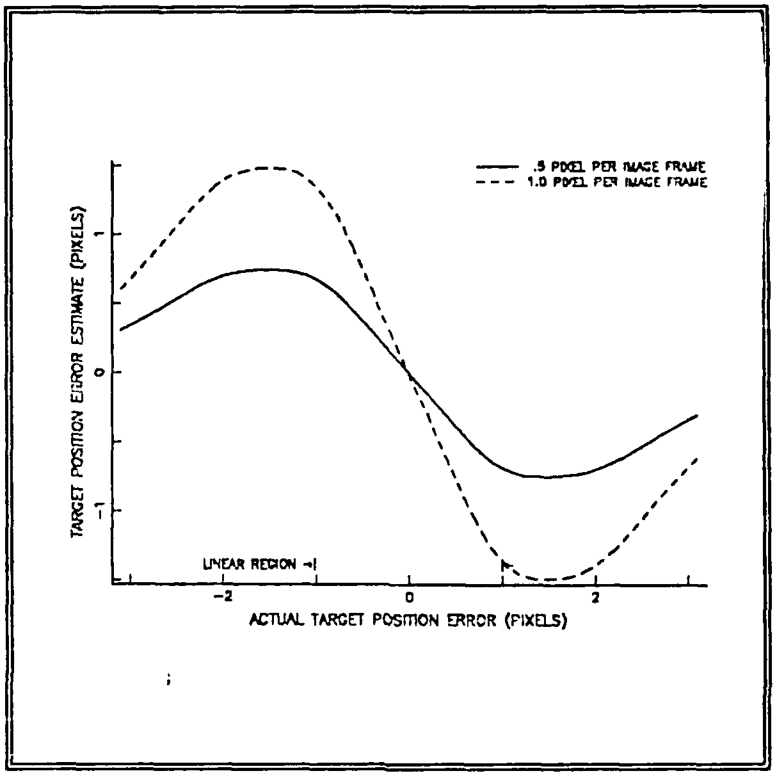

Figure 6-3. Estimated Position Error Versus Actual Position Error; σ_s^2=2.0, $r_z^n(0)$=.7 for .5 Pixel per Frame, $r_z^n(0)$=1.2 for 1.0 Pixel per Frame

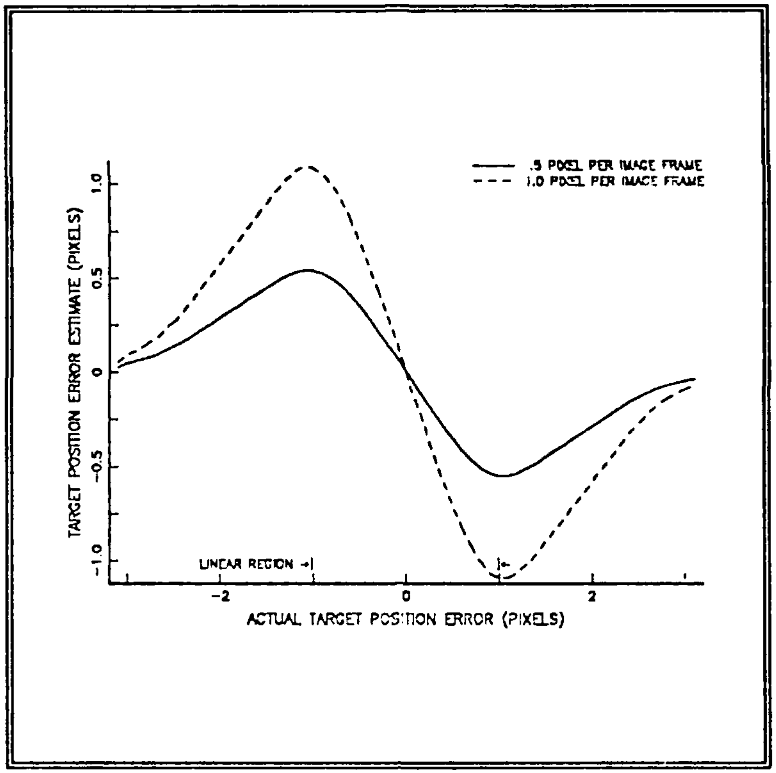

Figure 6-4. Estimated Position Error Versus Actual Position Error (Target Track 45 Degrees); σ_s^2=2.0, $r_z^n(0)$=.7 for .5 Pixel per Frame, $r_z^n(0)$=.5 for 1.0 Pixel per Frame

for target motion at 45 degrees away from the horizontal with a horizontal component of .5 or 1.0 pixels per image. In both plots it can be seen that the estimator response to the actual position error is divided into two regions. The first is a linear region centered at the origin and extending for about ±1 pixel in true position error. The remainder is a region where the response is nonlinear.

For these plots the term $r_z^n(0)$ has been adjusted to place the known maximum displacement between frames in the linear region of the estimator response. The region beyond this point is the nonlinear region. In this region the estimates of the target position error do not match the actual target position error. Thus, the result of setting the position error variance to match a specific target motion is that those with greater motion will tend to be rejected.

It is apparent from the plots that only a small portion of the tracking window {-3,3} is contained in the linear region. This is primarily a result of the gradient of the target model. Error contributions between the modeled target and the observed target (6-33) receive considerably less weight as the distance from the centroid increases. A possible means of extending the effective area of the estimator is through the use of a constant magnitude gradient. The form of the estimators for the "constant gradient" method is

$$\Delta_n{}^{p+1} = \frac{1}{N_0 W^2} \sum_{n=-\frac{W+1}{2}}^{\frac{W-1}{2}} \sum_{m=-\frac{W+1}{2}}^{\frac{W-1}{2}} -2\mathrm{sgn}(n)\underline{r}_z^n(\underline{r}-\underline{s}(\zeta)) \tag{6-36}$$

and

$$\Delta_m{}^{p+1} = \frac{1}{N_0 W^2} \sum_{n=-\frac{W+1}{2}}^{\frac{W-1}{2}} \sum_{m=-\frac{W+1}{2}}^{\frac{W-1}{2}} -2\mathrm{sgn}(m)\underline{r}_z^m(\underline{r}-\underline{s}(\zeta)) \tag{6-37}$$

where the gradient in (6-32) and (6-33) has been replaced by -2sgn(n) and -2sgn(m). Plots of the estimator response with the constant gradient are shown in Figures 6-5 and 6-6. The value of r_x was decreased to produce estimators with the same power as those in the previous plots. The target motion in Figures 6-5 and 6-6 is the same as that in Figures 6-3 and 6-4 respectively.

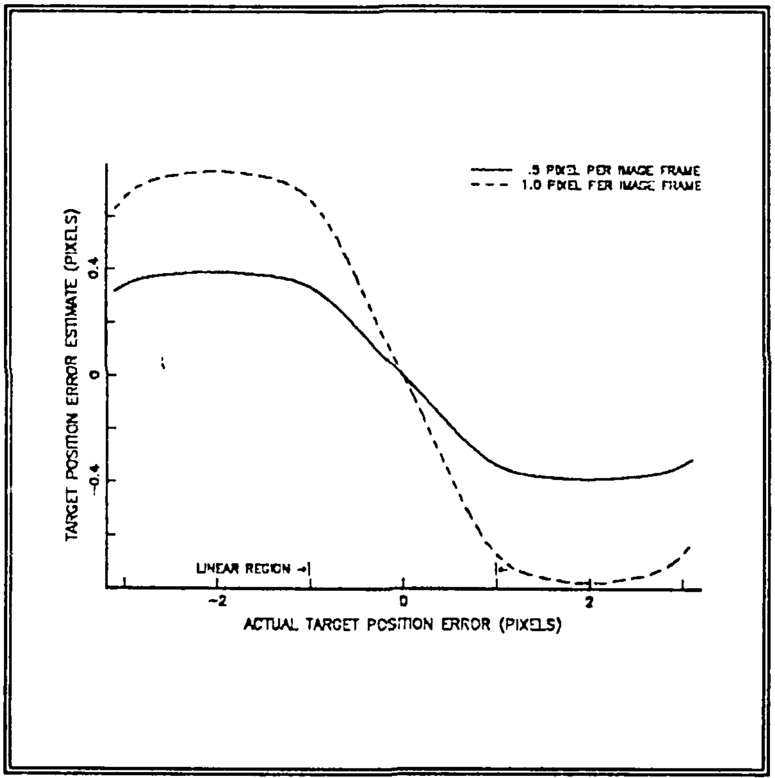

Figure 6-5. Estimated Position Error Versus Actual Position Error; σ_s^2=2.0, $r_z^n(0)$=.1 for .5 Pixel per Frame, $r_z^n(0)$=.2 for 1.0 Pixel per Frame

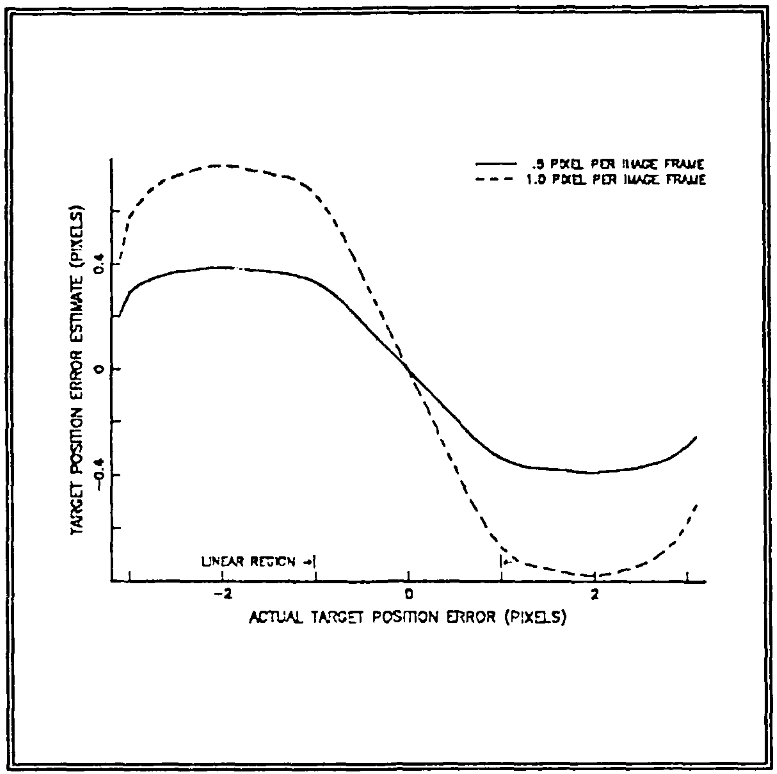

Figure 6-6. Estimated Position Error Versus Actual Position Error (Target Track 45 Degrees); σ_s^2=2.0, r_z^n(0)=.1 for .5 Pixel per Frame, r_z^n=.2 for 1.0 Pixel per Frame

It can be seen that using a constant gradient flattens the response of the tracker in the nonlinear region. The effect of maintaining the same power is to reduce in the magnitude of the estimate near the origin. However, this is offset by the increased magnitude of the estimates further from the origin. Thus, the effective response region has been increased virtually to the window limit (-3,3).

D. IMPLEMENTATION OF THE TRACKER

After the images are processed by the target detection algorithm the resulting targets and prediction error image are passed to the tracker. The targets' locations are placed in the tracker's "Track File" until retreived by the tracking window control unit. After retrieval of the target data the tracking window control unit accesses the prediction error image (see Figure 6-7). The data within the computed 7x7 tracking window, centered on the designated target coordinates, is extracted.

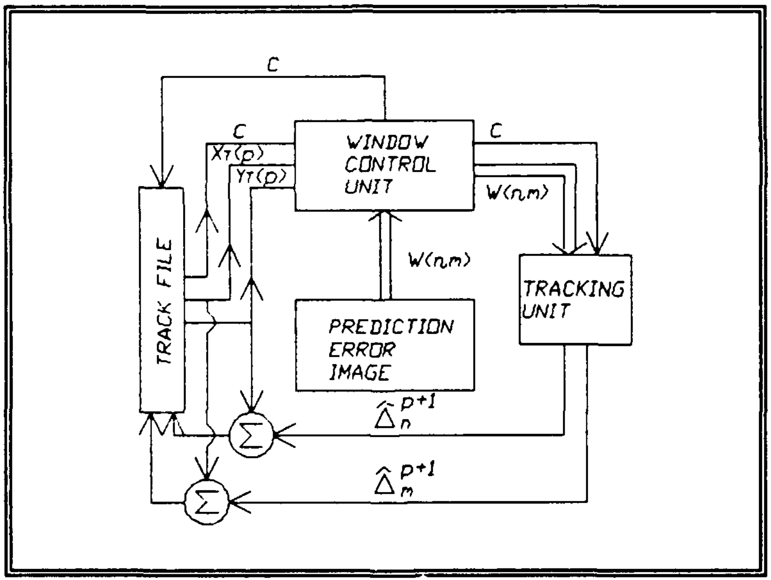

Figure 6-7. Tracker and Trackfile

At this point an estimate of the polarity and magnitude of the target is computed by

$$C = e(0,0) - \frac{1}{46} \sum_{n=-3}^{3} \sum_{\substack{m=-3 \\ (n,m)\neq\{(0,0)(0,1)(1,0)\}}}^{3} e(n,m) \qquad (6\text{-}38)$$

where e(n,m) is the prediction error residue within the tracking window. The term e(0,0) is the target intensity, located at the tracking window centroid. To compute the estimate the intensities of pixels considerably affected by the impulse response due to the target have been excluded. As mentioned previously these pixels are generally within one pixel of the target.

After this computation, both the tracking window data array and the value of the scaling factor C are passed to the tracking unit. The tracking unit computes an estimate of the target position error using (6-32) and (6-33) or (6-36) and (6-37). The estimate is passed to the tracking window control unit which computes an updated tracking window centroid and repositions the tracking window. This new position and the estimated value of C are stored in the track file until a new prediction error image becomes available and the track file has been completely processed.

A block diagram of the tracking unit is shown in Figure 6-8. The summer computes the difference between the tracking window array and the target model array and the output is multiplied by the reciprocal of the estimate for the noise

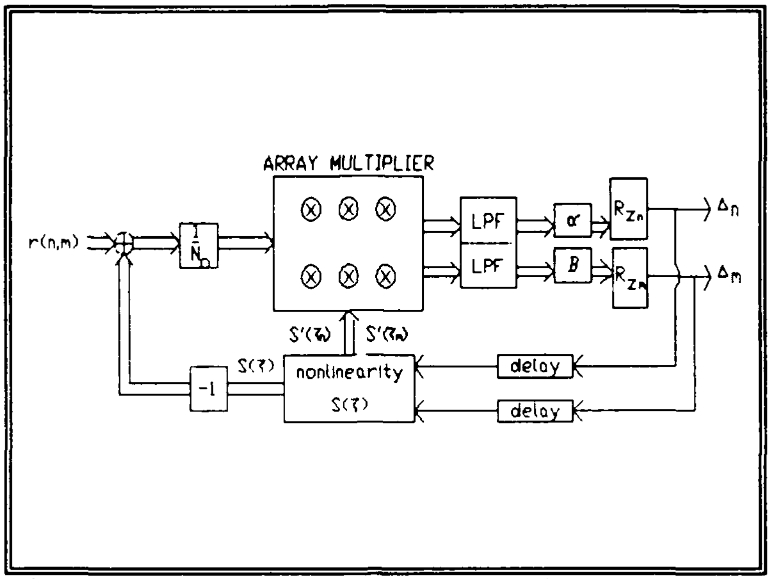

Figure 6-8. Tracking Unit Diagram

power. Two array multipliers are used to multiply this result by the target model gradient with respect to the n and m directions. For the general method represented by (6-26) and (6-27) the output of the array multiplier is averaged and the result is stored in a stack. The averaging is accomplished by the block designated as LPF. The oldest values of $\alpha_{avg}(i)$ and $\beta_{avg}(i)$ are shifted out of the stack and discarded. The contents of the α_{avg} and β_{avg} buffers are vector multiplied by the k lags of the position error autocorrelation function. The result is the updated estimate of the position error. For uncorrelated target motion represented by (6-32) and (6-33) use of the stack is not required.

Gradient and modeled target intensity values for locations within the tracking window are provided to both the summer and the array multiplier by the block labeled "Nonlinearity $\mathbf{S}(\zeta)$." This block is implemented as a look-up table.

The system is initialized with the initial target coordinates, the previously estimated C term, and by setting $\alpha_{avg}(i)=\beta_{avg}(i)=0$ for all lags.

E. TEST RESULTS

The following section is comprised of tracking runs at a variety of noise levels and for several different trajectories. All results were produced with the same uncorrelated trajectory model (i.e., only the zero-th lag of $\underline{r}_z$ was not equal to zero). The results are presented in Table 6-1 and in Figures 6-9 through 6-20. The table contains, for each figure, the signal to noise ratio, the estimated variance of the position error $r_z(0)$, and the resulting sample error variance (Var(n) and Var(m)) in the estimates.

In the first trajectory (Figures 6-9 through 6-14) the target motion was .5 pixels per image frame in both the row and column direction. In the second trajectory (Figures 6-15 through 6-20) the target was moving at approximately 1 pixel per frame. All figures in the simulations show motion through 20 image frames. This trajectory started at the lower

TABLE 6-1. TRACKING SIMULATION RESULTS

Figure #	SNR(dB)	$r_z(0)$	Var(n)	Var(m)
6-9(p)[3]	6.4	0.7	.075	.075
6-10(p)	2.8	0.7	.114	.134
6-11(p)	1.2	0.7	.293	.147
6-12(c)[4]	6.4	0.1	.388	.310
6-13(c)	2.8	0.1	.498	.383
6-14(c)	1.2	0.1	.558	.190
6-15(p)	6.4	1.2	.071	.063
6-16(p)	2.8	1.2	.245	.418
6-17(p)	1.2	1.2	.323	.277
6-18(c)	6.4	0.2	.355	.316
6-19(c)	2.8	0.2	.513	.698
6-20(c)	1.2	0.2	.753	.344

[3] p indicates that the pulse gradient method was used.

[4] c indicates that the constant gradient method was used.

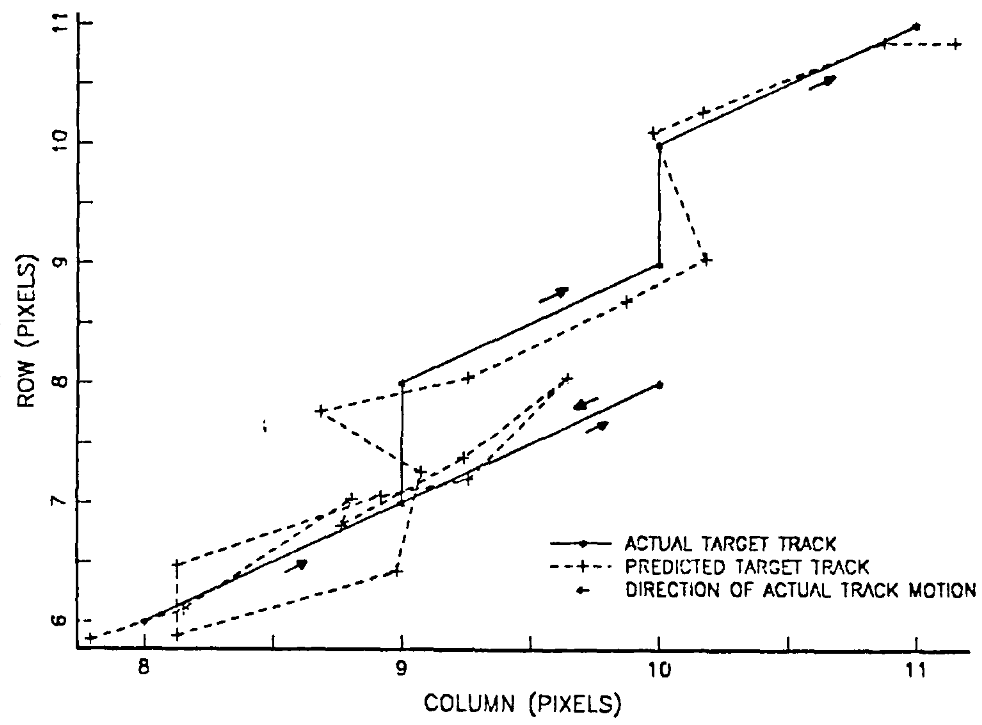

Figure 6-9. Pulse Gradient Model Track SNR=6.4dB

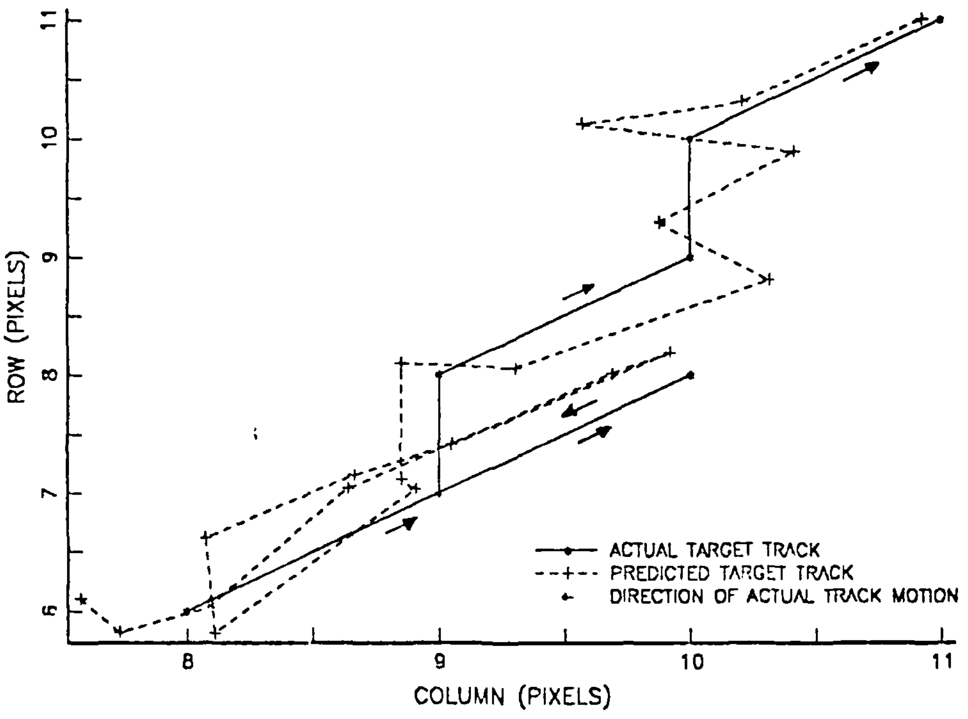

Figure 6-10. Pulse Gradient Model Track SNR=2.8dB

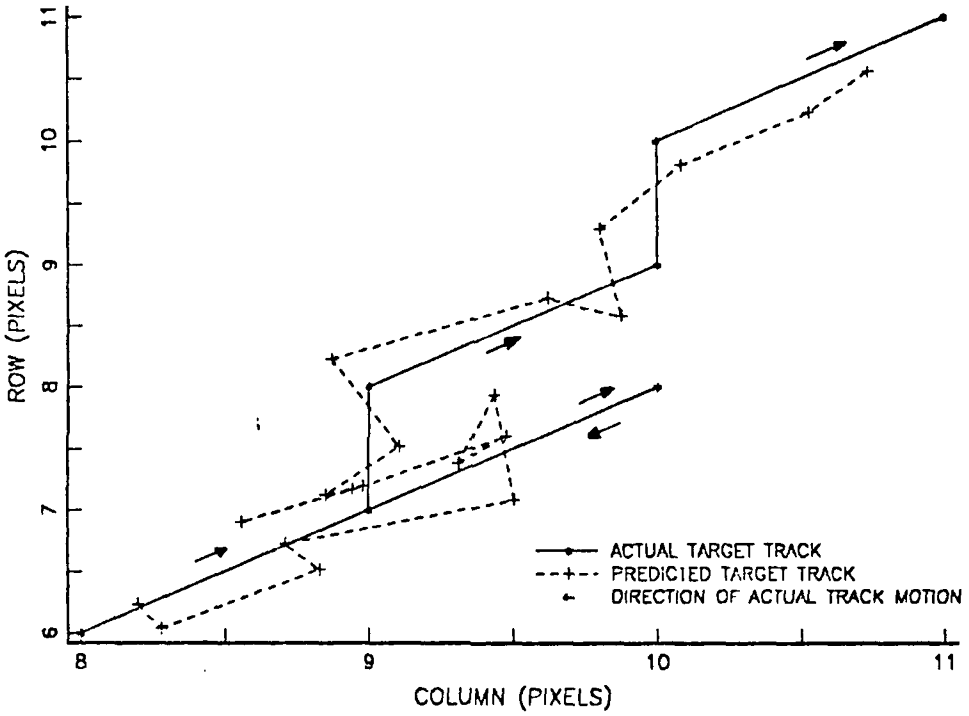

Figure 6-11. Pulse Gradient Track SNR=1.2dB

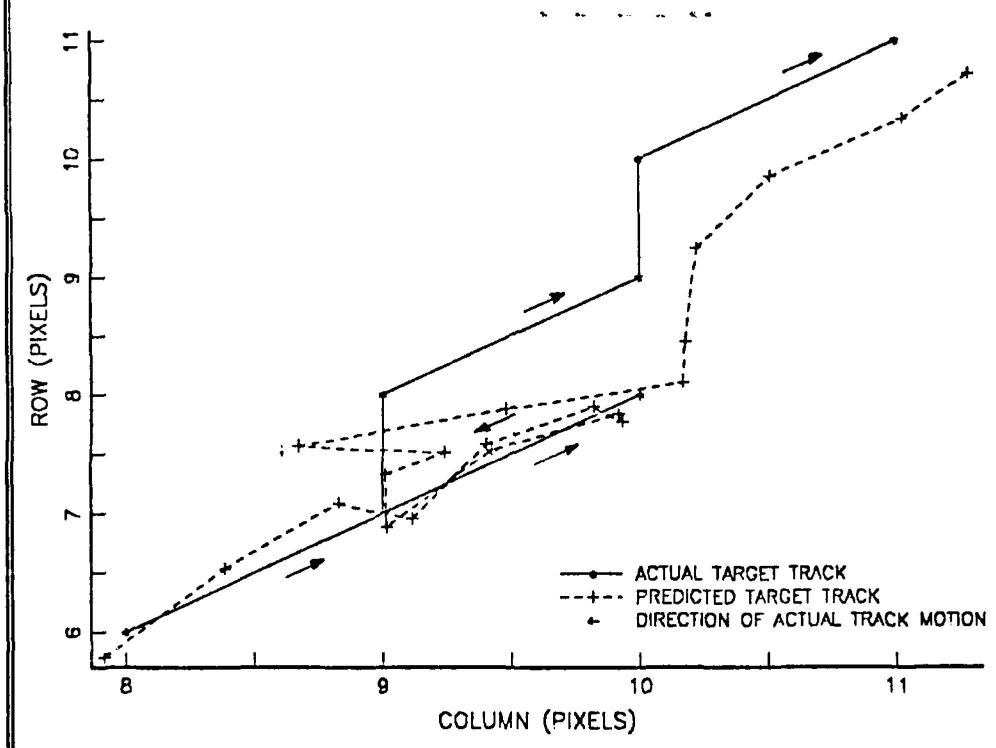

Figure 6-12. Constant Gradient Track SNR=6.4dB

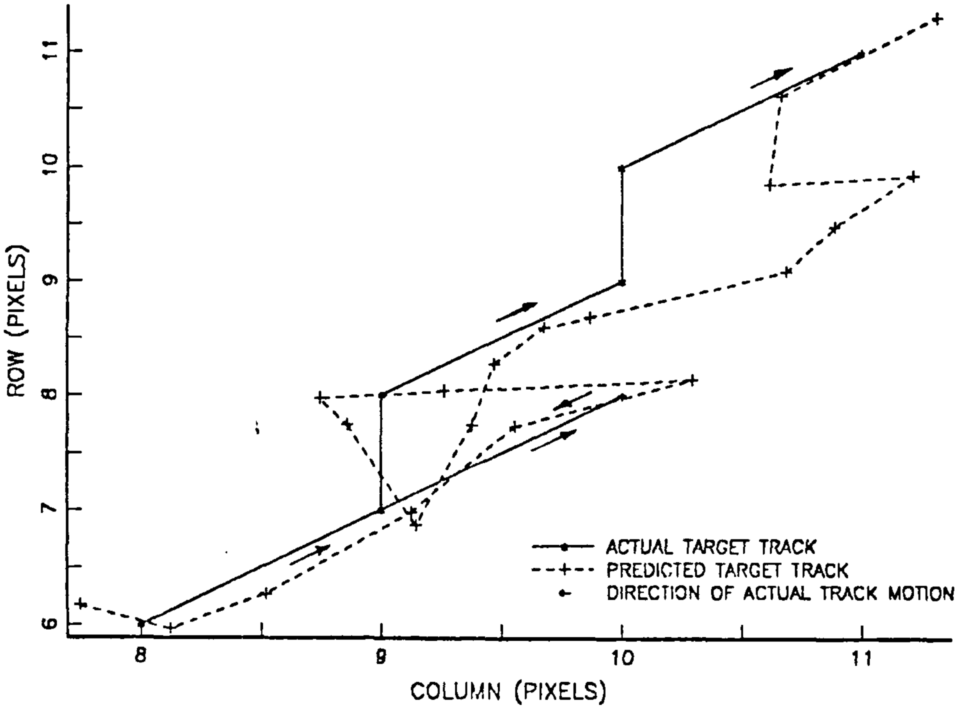

Figure 6-13. Constant Gradient Track SNR=2.8dB

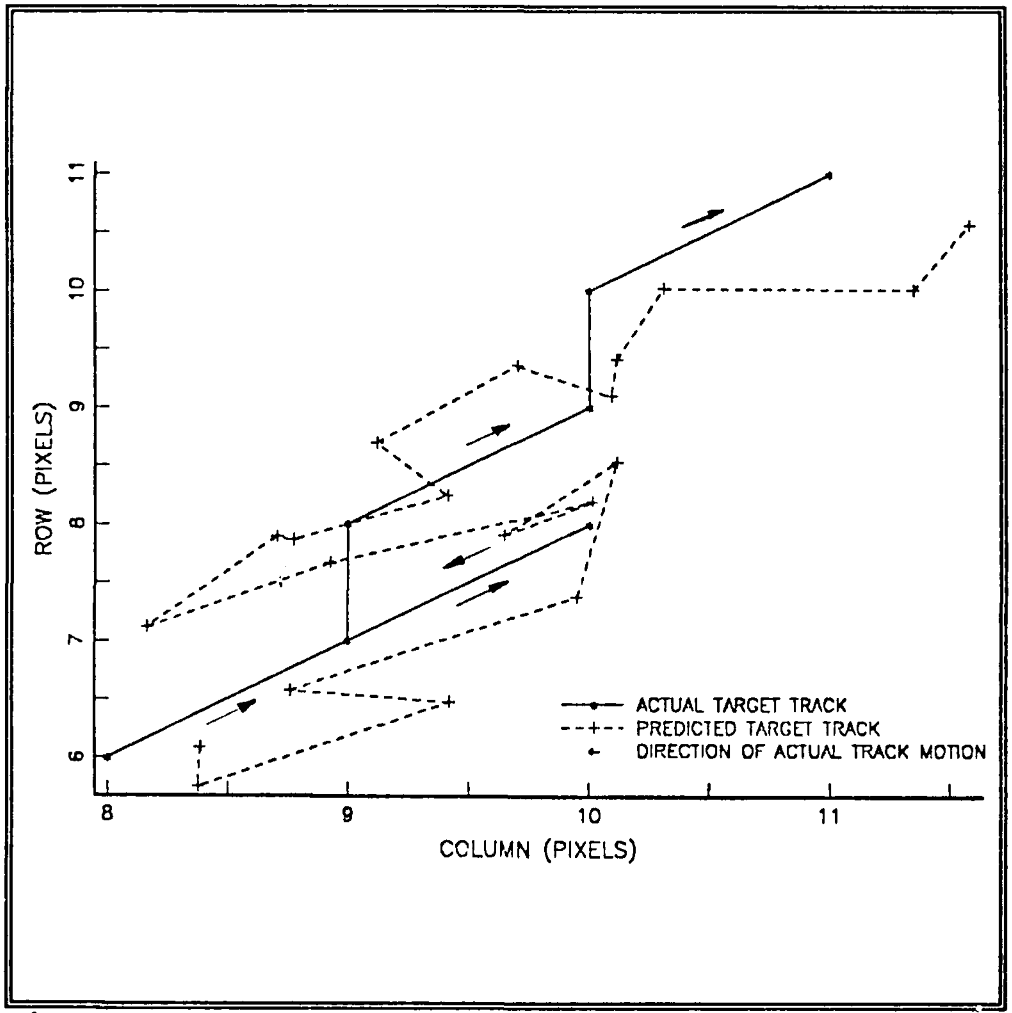

Figure 6-14. Constant Gradient Track SNR=1.2dB

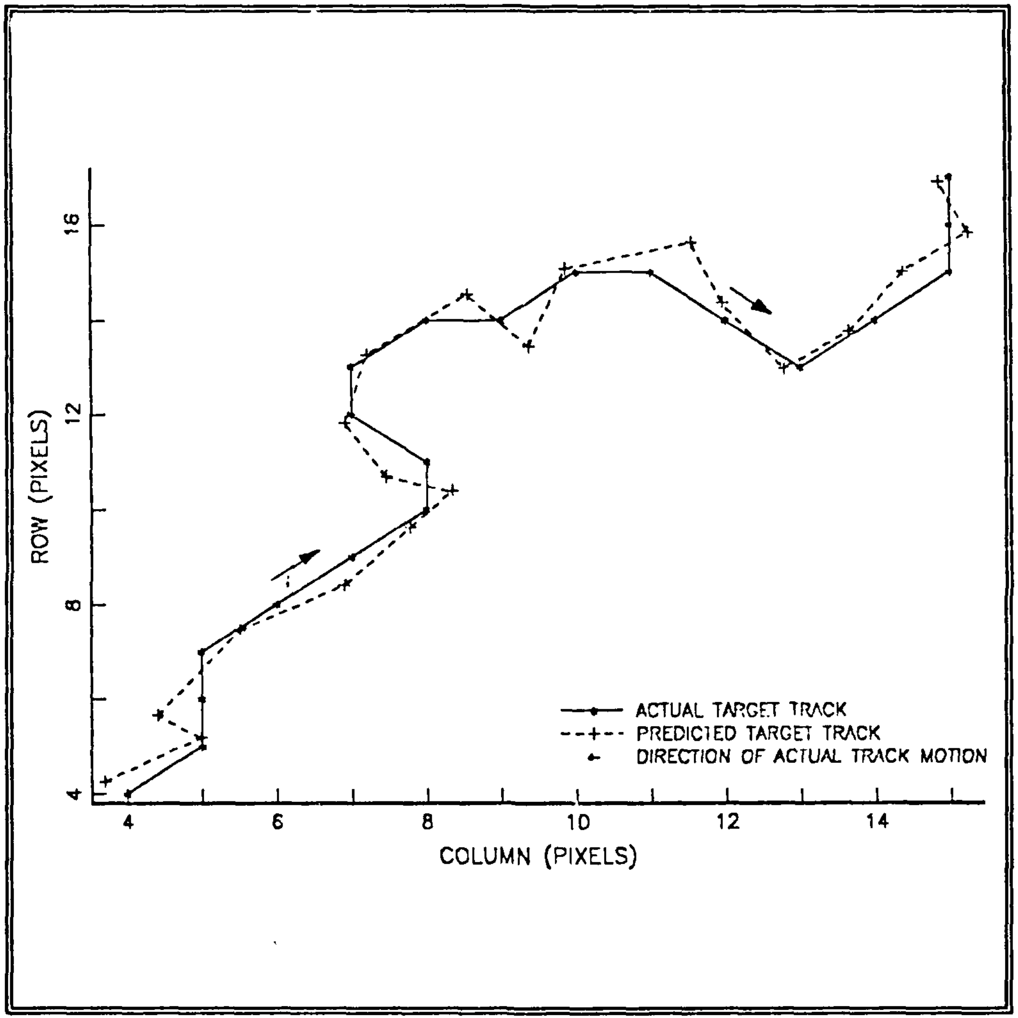

Figure 6-15. Pulse Gradient Track SNR=6.8dB

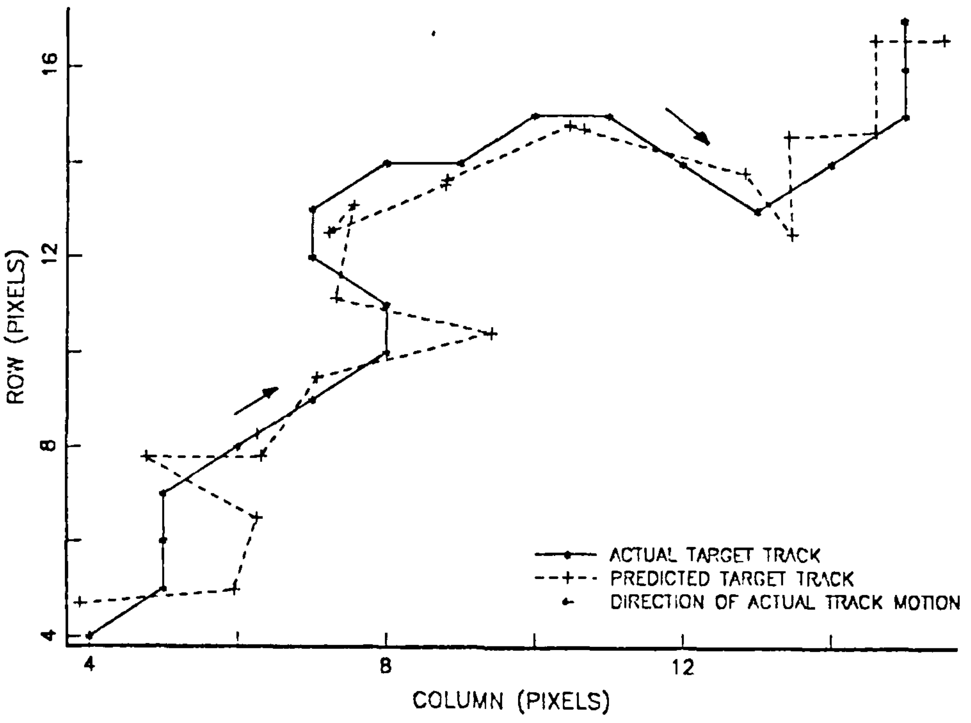

Figure 6-16. Pulse Gradient Track SNR=2.8dB

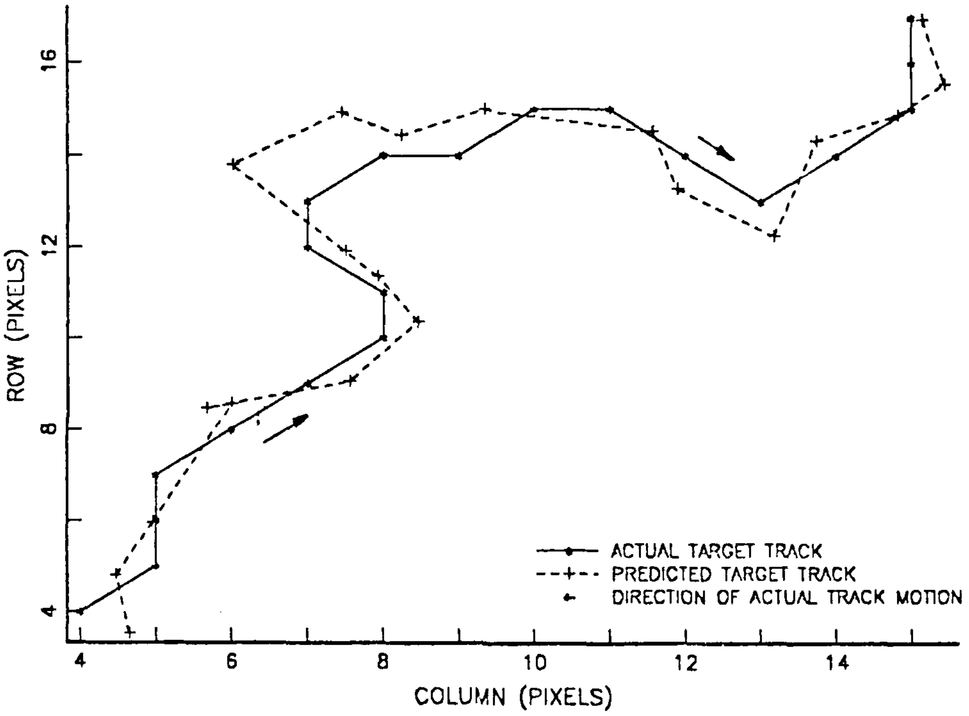

Figure 6-17. Pulse Gradient Track SNR=1.2dB

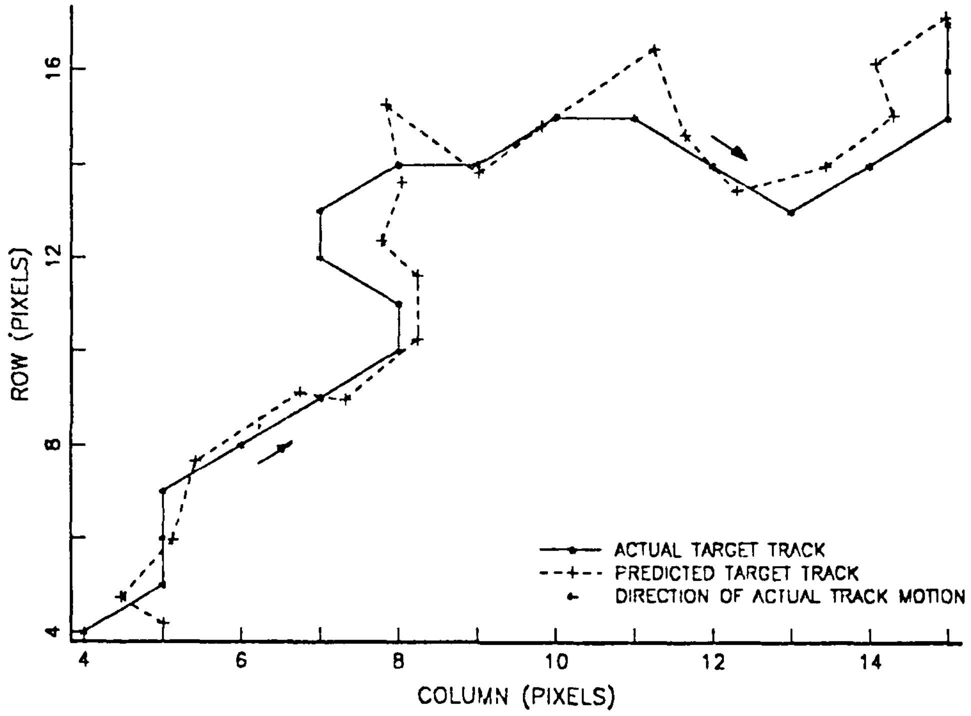

Figure 6-18. Constant Gradient Track SNR=6.8dB

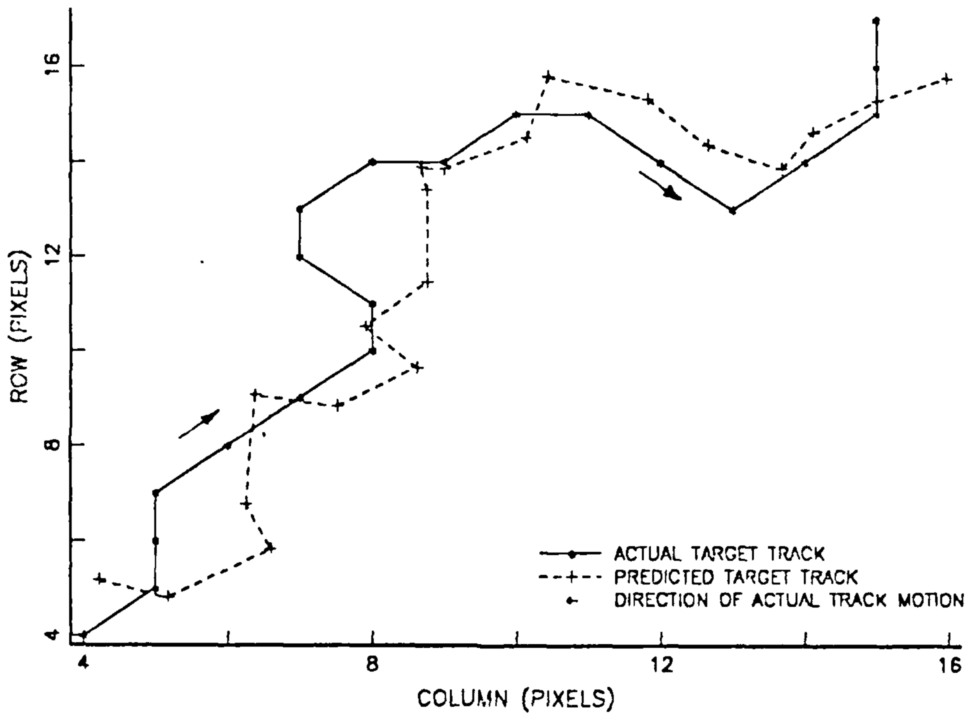

Figure 6-19. Constant Gradient Track SNR=2.8dB

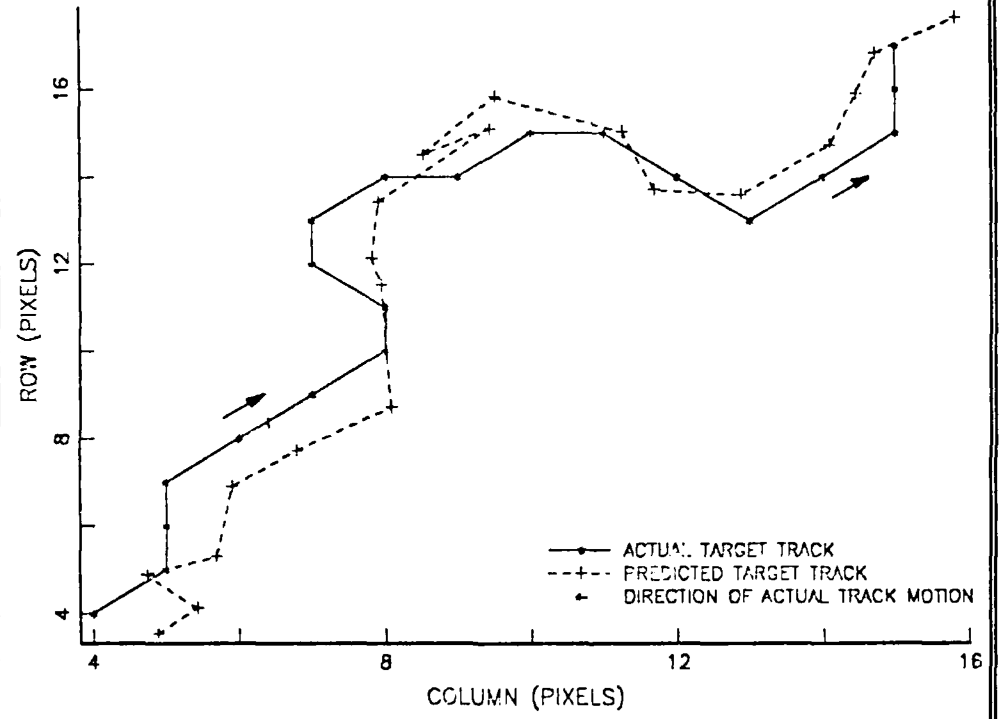

Figure 6-20. Pulse Gradient Track SNR=1.2dB

left corner and moved to the upper right corner. In Figures 6-9 through 6-14 a 180 degree direction reversal was made by the target, at the location (10,8), followed later by other abrupt changes in the trajectory. The figures show that the estimated trajectory follows the track well at all noise levels. The table further shows that the computed error variances, Var(n) and Var(m) are consistently less than one pixel.

The estimates of the position error were computed using (6-3), (6-32), and (6-33) evaluated at the discrete points in the tracking window. This involved floating point operations and round off of the final position error estimate to the nearest integer value. As a result, quantization noise becomes a factor in the position error estimates. The quantization level separation in the case of an image is one pixel. Therefore assuming a uniform probability distribution for the floating point result, a quantization noise variance of .083 pixel^2 is expected. It can be seen for the higher signal-to-noise level that the results are close to the quantization level.

The noise rejection capability of the tracking algorithm using the pulse gradient form is seen to be high. In Figures 6-9 through 6-11 and 6-15 through 6-17 the estimated tracks remain within one pixel of the true track. However, as might be expected, the constant gradient method is more susceptible to noise and shows deviations from the track exceeding a pixel at even the higher signal-to-noise levels (see Figures 6-12 through 6-14 and 6-18 through 6-20). However it may be required to use the constant gradient method if the target motion exceeds 2 pixels per image frame. When the target shift between frames exceeds the width of the window, the track will be lost. Simulations have shown that if only noise is contained in the target window the track will stagnate or drift very little. The effective radius for the pulse gradient model was found to be approximately $1.5\sigma_s$, where σ_s is the target model shape factor. This can be tied to the decreased magnitude of the gradient beyond this value. For the constant gradient method, the effective radius is essentially equal to the window dimension (W). However, because of this, the constant gradient method is more susceptible to clutter or targets crossing the boundaries of the tracking window.

F. CONCLUSIONS

This chapter was divided into several major areas. The first was a general description of the interface between the detection algorithm and the tracking algorithm. The other areas consisted of a development of the tracker equations, a dynamic analysis of the tracker, and simulation results.

The results of these and additional simulations verified the tracker's expected performance. It was found through simulation that the gradient of the proposed target model limits the effective rate at which targets can move and still be tracked. This prompted consideration of the constant gradient model. In this model the modeled target intensity was left unchanged but the gradient was made a constant with appropriate sign. Use of the constant gradient was found to increase the track noise and decrease the resolution (i.e., minimum spacing between adjacent tracks) of the tracker.

Either form of the tracker could be used as a target dynamics observer. An observer is a system which can produce estimates of parameters (e.g., velocity) that are not readily available. This observer information could be incorporated into the overall track file reduction process proposed earlier. This may be a profitable area for future research.

VII. CONCLUSIONS AND RECOMMENDATIONS

A. GOALS

The purpose of this thesis was to develop algorithms for the detection and tracking of dim point targets in infrared images. Images of this type include those produced by satellite-mounted infrared sensors. The targets are restricted to an approximate size of one image pixel and typically move at rates of approximately 1 to 2 pixels between successive image frames. A goal was also to develop these algorithms so they would be easy to implement and apply to other images in other spectral bands.

B. SOLUTION

The solution presented involves three steps. The first is to filter the image with an adaptive two-dimensional fixed size prediction error filter (PEF). This has the effect of whitening the image background and producing spikes at points, such as targets, different from the background. Both two-dimensional Least Mean Square (LMS) and 2-D Recursive Least Squares (RLS) filters were used. Both of these PEFs proved effective in whitening the background processes, however, the RLS method proved to be superior to the LMS method in the low signal-to-noise ratio regime. This was shown to be important in the next step of processing.

The second step involves significance testing the error residual produced by inverse filtering. This results in suppression of the error residual produced by inverse filtering the background and highlighting the targets. The results presented showed that the LMS method could achieve a 90% detection probability for 0 dB targets in the presence of light noise (30 dB SNR) with less than 9% false alarms. The RLS method produced comparable results for the same parameters. However, with moderate noise of (10 dB SNR), a 2 dB target and a 90% detection probability, the LMS false alarm rate increased to 15% while that of the RLS increased only to 10%. The performance of the RLS detection algorithm was significantly improved at target levels of approximately 1 to 2 dB. In light noise this resulted in a 90% detection rate for an approximately 5% false alarm rate. Both detection methods were found to be easy to implement and relatively inexpensive.

The final step taken in the problem solution was the development of a target tracker. This tracker receives initial target location information from the previous detection algorithm and tracks targets through successive frames of data. It was shown that a track can be maintained with less than one pixel error with signal-to-noise ratios near 0dB and for movement of 1 pixel per image frame. Additional simulations showed that track could also be maintained for 2 pixels per frame motion with approximately one pixel track error. The tracker was found to be robust with respect to the estimates of the required parameters and to the general assumptions made in the development.

C. RECOMMENDATIONS

The previously mentioned processing involves images from only one focal plane. Often there are more than one focal plane, in different spectral bands, available in the infrared sensors. The significance of a target in one band will differ from that in the other bands. This will in turn result in a different PDF for the target error process. If the results of significance testing multiple images in different bands can be merged it may be possible to considerably reduce the false alarm rate. This is suggested as a possible area for further research.

In order to implement the proposed method additional work needs to be done on the track file management logic. Some of the possible approaches and considerations were mentioned in Chapter VI Section A.

APPENDIX

This appendix contains the results of the detection process using the LMS and RLS, algorithms. A detailed discussion is presented in Chapter V.

TABLE 1. TEST RESULTS LMS PROCESSED DATA (NO NOISE)

BNR(dB)	TBR(dB)	Pf[5]	Pd[6]	THRESHOLD
na	0	.0022	.10	23.33
na	0	.0036	.20	14.16
na	0	.0056	.30	11.96
na	0	.0206	.40	6.64
na	0	.0405	.50	4.94
na	0	.0569	.60	3.96
na	0	.0647	.70	3.63
na	0	.0755	.80	3.28
na	0	.1437	.90	2.14
na	0	.1826	1.00	1.67

[5] Pf is the ratio of the sum of non-target pixels above the threshold to the total number of pixels tested.

[6] Pd is the ratio of the sum of target pixels above the threshold to the total number of targets present.

TABLE 2. TEST RESULTS FOR LMS PROCESSED DATA

BNR(dB)	TBR(dB)	Pf	Pd	THRESHOLD
30	0	.0024	.10	20.47
30	0	.0029	.20	16.03
30	0	.0066	.30	9.97
30	0	.0085	.40	8.77
30	0	.0135	.50	7.53
30	0	.0372	.60	5.31
30	0	.0455	.70	4.73
30	0	.0471	.80	4.63
30	0	.0827	.90	3.08
30	0	.1145	1.00	2.45

TABLE 3. TEST RESULTS LMS PROCESSED DATA (NO NOISE)

BNR(dB)	TBR(dB)	Pf	Pd	THRESHOLD
na	2	.0022	.10	22.98
na	2	.0031	.20	14.07
na	2	.0044	.30	11.15
na	2	.0106	.40	8.35
na	2	.0241	.50	6.29
na	2	.0371	.60	5.01
na	2	.0608	.70	3.94
na	2	.0668	.80	3.63
na	2	.1116	.90	2.62
na	2	.1772	1.00	1.72

TABLE 4. TEST RESULTS LMS PROCESSED DATA

BNR(dB)	TBR(dB)	Pf	Pd	THRESHOLD
30	2	.0016	.10	28.36
30	2	.0024	.20	19.20
30	2	.0033	.30	12.99
30	2	.0042	.40	11.33
30	2	.0113	.50	8.10
30	2	.0215	.60	6.56
30	2	.0347	.70	5.65
30	2	.0372	.80	5.44
30	2	.0699	.90	3.78
30	2	.1091	1.00	2.76

TABLE 5. TEST RESULTS LMS PROCESSED DATA

BNR(dB)	TBR(dB)	Pf	Pd	THRESHOLD
10	2	.0052	.10	17.65
10	2	.0079	.20	11.18
10	2	.0125	.30	8.65
10	2	.0213	.40	6.74
10	2	.0266	.50	5.72
10	2	.0412	.60	4.90
10	2	.0545	.70	4.24
10	2	.0892	.80	3.12
10	2	.1557	.90	2.24
10	2	.2491	1.00	1.42

TABLE 6. TEST RESULTS LMS PROCESSED DATA
(NO NOISE)

BNR(dB)	TBR(dB)	Pf	Pd	THRESHOLD
na	5	.0014	.10	38.09
na	5	.0020	.20	24.77
na	5	.0024	.30	19.31
na	5	.0031	.40	13.31
na	5	.0040	.50	11.43
na	5	.0132	.60	7.78
na	5	.0269	.70	5.70
na	5	.0472	.80	4.32
na	5	.0621	.90	3.70
na	5	.1529	1.00	2.03

TABLE 7. TEST RESULTS LMS PROCESSED DATA

BNR(dB)	TBR(dB)	Pf	Pd	THRESHOLD
30	5	.0016	.10	38.44
30	5	.0023	.20	22.79
30	5	.0029	.30	16.37
30	5	.0033	.40	13.78
30	5	.0042	.50	11.45
30	5	.0104	.60	8.17
30	5	.0169	.70	7.09
30	5	.0259	.80	6.19
30	5	.0622	.90	4.02
30	5	.1386	1.00	2.33

TABLE 8. TEST RESULTS LMS PROCESSED DATA

BNR(dB)	TBR(dB)	Pf	Pd	THRESHOLD
10	5	.0016	.10	29.35
10	5	.0022	.20	21.98
10	5	.0031	.30	14.41
10	5	.0046	.40	12.24
10	5	.0182	.50	7.55
10	5	.0272	.60	6.51
10	5	.0500	.70	4.23
10	5	.0655	.80	3.65
10	5	.0835	.90	3.19
10	5	.2229	1.00	1.79

TABLE 9. TEST RESULTS RLS PROCESSED DATA (NO NOISE)

BNR(dB)	TBR(dB)	Pf	Pd	THRESHOLD
na	0	0	.10	22.99
na	0	.0005	.20	15.06
na	0	.0013	.30	13.21
na	0	.0071	.40	9.74
na	0	.0107	.50	8.42
na	0	.0151	.60	7.59
na	0	.0248	.70	6.70
na	0	.0468	.80	5.20
na	0	.0887	.90	3.44
na	0	.1214	1.00	2.80

TABLE 10. TEST RESULTS RLS PROCESSED DATA

BNR(dB)	TBR(dB)	Pf	Pd	THRESHOLD
30	0	0	.10	22.68
30	0	.0004	.20	15.57
30	0	.0015	.30	13.40
30	0	.0076	.40	9.51
30	0	.0112	.50	8.27
30	0	.0164	.60	7.44
30	0	.0258	.70	6.57
30	0	.0513	.80	4.90
30	0	.0822	.90	3.63
30	0	.1169	1.00	2.88

TABLE 11. TEST RESULTS RLS PROCESSED DATA
(NO NOISE)

BNR(dB)	TBR(dB)	Pf	Pd	THRESHOLD
na	2	0	.10	27.66
na	2	.0001	.20	18.45
na	2	.0003	.30	16.34
na	2	.0035	.40	11.38
na	2	.0054	.50	10.22
na	2	.0066	.60	9.64
na	2	.0133	.70	7.77
na	2	.0234	.80	6.71
na	2	.0579	.90	4.47
na	2	.1126	1.00	3.03

TABLE 12. TEST RESULTS RLS PROCESSED DATA

BNR(dB)	TBR(dB)	Pf	Pd	THRESHOLD
30	2	0	.10	27.28
30	2	0	.20	18.47
30	2	.0005	.30	16.25
30	2	.0034	.40	11.51
30	2	.0052	.50	10.00
30	2	.0080	.60	9.27
30	2	.0153	.70	7.39
30	2	.0252	.80	6.46
30	2	.0555	.90	4.69
30	2	.1208	1.00	2.95

TABLE 13. TEST RESULTS RLS PROCESSED DATA

BNR(dB)	TBR(dB)	Pf	Pd	THRESHOLD
10	2	.0005	.10	16.39
10	2	.0051	.20	11.12
10	2	.0120	.30	8.21
10	2	.0255	.40	6.50
10	2	.0427	.50	5.39
10	2	.0519	.60	5.02
10	2	.0711	.70	4.33
10	2	.0917	.80	3.88
10	2	.1085	.90	3.56
10	2	.2157	1.00	2.35

TABLE 14. TEST RESULTS RLS PROCESSED DATA (NO NOISE)

BNR(dB)	TBR(dB)	Pf	Pd	THRESHOLD
na	5	0	.10	36.24
na	5	0	.20	23.14
na	5	0	.30	20.65
na	5	.0002	.40	15.33
na	5	.0008	.50	12.49
na	5	.0038	.60	11.10
na	5	.0045	.70	10.31
na	5	.0141	.80	7.66
na	5	.0257	.90	6.26
na	5	.0604	1.00	4.49

TABLE 15. TEST RESULTS RLS PROCESSED DATA

BNR(dB)	TBR(dB)	Pf	Pd	THRESHOLD
30	5	0	.10	34.22
30	5	0	.20	24.77
30	5	0	.30	22.28
30	5	.0001	.40	17.28
30	5	.0005	.50	14.62
30	5	.0036	.60	13.19
30	5	.0044	.70	11.08
30	5	.0131	.80	8.15
30	5	.0267	.90	6.15
30	5	.0682	1.00	4.08

TABLE 16. TEST RESULTS RLS PROCESSED DATA

BNR(dB)	TBR(dB)	Pf	Pd	THRESHOLD
10	5	0	.10	23.42
10	5	0	.20	18.25
10	5	.0002	.30	15.52
10	5	.0007	.40	13.34
10	5	.0023	.50	11.46
10	5	.0062	.60	9.53
10	5	.0314	.70	6.99
10	5	.0409	.80	5.94
10	5	.0726	.90	5.16
10	5	.1057	1.00	3.69

LIST OF REFERENCES

1. Henderson, C. M., "A Detection and Processing Scheme for Moving Target Detection Using a Passive Sensor," AGARD Conference Proc., n. 5, pp. 21.1-21.9, November 1980.

2. Maybeck, P. S., and Rodgers, S. K., "Adaptive Tracking of Multiple Hot-Spot Target IR Images," IEEE Trans. on Automatic Control, v. 28, n. 10, pp. 937-943, October 1983.

3. Markham, B. L., "The Landstat Sensors' Spatial Responses," IEEE Trans. on Geosciences and Remote Sensing, v. 23, n. 6, pp. 864-875, November 1985.

4. Rauch, H. E., Futterman, W. I., and Kemmer, D. B., "Background Suppression and Tracking With a Staring Mosaic Sensor," Optical Engineering, v. 20, n. 1, pp. 103-110, January/February 1981.

5. Meyer, D., Muller, M., and Weimann, A., "Target Identification of Aircraft Using IR/TV-Sensor-Images," AGARD Conference Proc., n. 5, pp. 33.1-33.7, November 1980.

6. Therrien, C. W., Quatieri, T. F., and Dudgeon, D. E., "Statistical Model-Based Algorithms for Image Analysis," Proc. of the IEEE, v. 74, n. 4, April 1986.

7. Ekstrom, M. P., "Realizable Wiener Filtering in Two Dimension," IEEE Trans. on Acoustics, Speech and Signal Processing, v. 30, n. 1, pp. 31-40, February 1982.

8. Marzetta, T. L., "Two-Dimensional Linear Prediction: Auto Correlation Arrays, Minimum-Phase Prediction Error Filters, and Reflection Coefficient Arrays," IEEE Trans. on Acoustics, Speech, and Signal Processing, v. 28, pp. 725-733, December 1980.

9. Jain, A. K., "Advances in Mathematical Models for Image Processing," Proc. IEEE, v. 69, pp. 502-528, May 1981.

10. Lincoln Laboratory, Massachusetts Institute of Technology, Technical Report 632, Object Detection by Two-Dimensional Linear Prediction, by Quatieri, T. F., 28 January 1983.

11. Widrow, B., and others, "Stationary and Non-Stationary Learning Characteristics of the LMS Adaptive Filter," Proc. of the IEEE, v. 64, n. 8, pp. 1151-1161, August 1976.

12. Hadhound, M. M., and Thomas, D. W., "The Two-Dimensional Adaptive LMS (TDLMS) Algorithm," IEEE Trans. on Circuits and Systems, v. 35, n. 5, pp. 485-494, May 1988.

13. Jones, S. K., Calvin, R. K. III, and Reed, W. M., "Analysis of Error-Gradient Adaptive Linear Estimators for a Class of Stationary Dependent Processes," IEEE Trans. on Information Theory, v. 28, n. 2, pp. 318-329, March 1982.

14. Florian, S., and Arie, F., "Performance Analysis of the LMS Algorithm with a Tapped Delay Line (Two-Dimensional Case)," IEEE Trans. on Acoustics, Speech and Signal Processing, v. 34, n. 6, pp. 1542-1549, December 1986.

15. Davis, B. R., and Cowley, W. G., "Bias and Variance of Spectral Estimates from an All-Pole Digital Filter," IEEE Trans. on Acoustics, Speech, and Signal Processing, v. 30, n. 2, pp. 322-329, April 1982.

16. Maragos, P. A., Schafer, R. W., and Mersereau, R. M., "Two-Dimensional Linear Prediction and Its Application to Adaptive Predictive Coding of Images," IEEE Trans. on Acoustics, Speech, and Signal Processing, v. 32, n. 6, pp. 1213-1229, December 1984.

17. Treichler, J. R., "Adaptive Algorithms that Restore Signal Properties," School of Electrical Engineering, Cornell University, Ithaca, NY, May 1984.

18. Bitmead, R. R., and Anderson, B. D., "Performance of Adaptive Estimation Algorithms in Dependent Random Environments," IEEE Trans. on Automatic Controls, v. 25, n. 4, pp. 788-794, August 1980.

19. Hsia, T. C., "An Investigation of Adjustment Gain Design in Stationary and Non-stationary LMS Adaptive Algorithms," Presented at 6th IFAC Symp. on Identification and System Parameter Estimation, Washington, DC, 1982.

20. Eweda, E., and Macchi, O., "Tracking Error Bounds of Adaptive Non-stationary Filtering," Automatica, v. 21, n. 3, pp. 293-302, June 1985.

21. Macchi, O., "Optimization of Adaptive Identification for Time Varying Filters," IEEE Trans. on Automatic Controls, v. 31, pp. 283-287, March 1986.

22. Haykin, S., "Introduction to Adaptive Filters," pp. 129-161, MacMillan Publishing Co., 1984.

23. Haykin, S., "Adaptive Filter Theory," MacMillan Publishing Co., 1984.

24. Larson, H. J., "Introduction to Probability Theory and Statistical Inference," Third Edition, pp. 462-507, John Wiley and Sons, 1982.

25. Franklin, G. J., and Powell, J. D., "Digital Control of Dynamic Systems," pp. 217-234, Addison-Wesley Publishing Co., 1980.

26. Therrien, C. W., "Loop Structures for Tracking of Multidimensional Signals," Proc. of the IEEE International Symposium on Circuits and Systems, May 1983.

INITIAL DISTRIBUTION LIST

No. Copies

1. Defense Technical Information Center 2
Cameron Station
Alexandria, Virginia 22304-6145

2. Library, Code 0142 2
Naval Postgraduate School
Monterey, California 93943-5002

3. Chairman, Code 62 1
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, California 93943-5000

4. Charles W. Therrien, Code 62Ti 3
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, California 93943-5000

5. Murali Tummala, Code 62Tu 1
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, California 93943-5000

6. Lawerence J. Ziomek, Code 62Zm 1
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, California 93943-5000

7. George J. Thaler, Code 62Tr 1
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, California 93943-5000

8. Dr. Steve Yool, Code 743 1
Naval Ocean Systems Center
San Diego, California 92152-5000

9. Thomas V. DeMars, Jr. 2
2652 Santa Barbara Drive
Bethel Park, Pennsylvania 15102